UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI
MINISTERUL MUNCII,
FAMILIEI,
PROTECŢIEI SOCIALE ŞI
PERSOANELOR VÂRSTNICE
AMPOSDRU

Fondul Social
European
POSDRU 2007-2013

Instrumente
Structurale
2007-2013

MINISTERUL
EDUCAŢIEI
NAŢIONALE

OIPOSDRU

UNIVERSITATEA
TEHNICĂ DE
CONSTRUCŢII DIN
BUCUREŞTI

Daniela HOSSU

Ioana FĂGĂRĂŞAN
Nicoleta ARGHIRA

Iulia DUMITRU
Sergiu Stelian ILIESCU

CARTE UNIVERSITARĂ

GHID PRACTIC DE PROIECTARE ŞI IMPLEMENTARE A APLICAŢIILOR SCADA

Proiect cofinanțat din Fondul Social European prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007-2013
Investește în OAMENI

MINISTERUL
EDUCAŢIEI
NAŢIONALE

MINISTERUL
EDUCAŢIEI
NATIONALE

UNIVERSITATEA
"POLITEHNICA"
DIN
BUCUREŞTI

UNIVERSITATEA TEHNICĂ
DIN
CLUJ-NAPOCA

UNIVERSITATEA TEHNICĂ
"GHEORGHE ASACHI"
DIN
IAŞI

UNIVERSITATEA
"POLITEHNICA"
DIN
TIMIŞOARA

SC ASTI AUTOMATION SRL

CONSPRESS BUCUREŞTI

UNIUNEA EUROPEANĂ

GUVERNUL ROMÂNIEI
MINISTERUL MUNCII, FAMILIEI, PROTECŢIEI SOCIALE ŞI PERSOANELOR VÂRSTNICE
AMPOSDRU

Fondul Social European
POSDRU 2007-2013

Instrumente Structurale
2007-2013

MINISTERUL EDUCAŢIEI NAŢIONALE
OIPOSDRU

UNIVERSITATEA TEHNICĂ DE CONSTRUCŢII DIN BUCUREŞTI

Investeşte în oameni !
Proiect cofinanţat din FONDUL SOCIAL EUROPEAN
prin Programul Operaţional Sectorial Dezvoltarea Resurselor Umane 2007 – 2013
Axa prioritară 1: „EDUCAŢIA ŞI FORMAREA PROFESIONALĂ ÎN SPIJINUL CREŞTERII ECONOMICE ŞI DEZVOLTĂRII SOCIETĂŢII BAZATE PE CUNOAŞTERE"
Domeniul major de intervenţie 1.2 „Calitate în învăţământul superior"
Cerere de propuneri de proiecte: nr. 86 „Universitate pentru viitor"
Titlul proiectului: ***"Reţea naţională de centre pentru dezvoltarea programelor de studii cu rute flexibile şi a unor instrumente didactice la specializarea de licenţă şi masterat, din domeniul Ingineria Sistemelor"***
Numărul de identificare al contractului: **POSDRU/86/1.2/S/63806**

Cartea se adresează celor care studiază în domeniul Ingineria Sistemelor la diferite specializări, la toate Facultăţile partenere în proiect, dar nu numai.

Îndrumarul a fost editat în cadrul perioadei de implementare a proiectului **POSDRU/86/1.2/S/63806**

Proiect cofinanţat din Fondul Social European prin Programul Operaţional Sectorial Dezvoltarea Resurselor Umane 2007-2013 — Investeşte în OAMENI

MINISTERUL EDUCAŢIEI NAŢIONALE

MINISTERUL EDUCAŢIEI NATIONALE

UNIVERSITATEA "POLITEHNICA" DIN BUCUREŞTI

UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA

UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" DIN IAŞI

UNIVERSITATEA "POLITEHNICA" DIN TIMIŞOARA

SC ASTI AUTOMATION SRL

Daniela HOSSU
Ioana FĂGĂRĂŞAN **Iulia DUMITRU**
Nicoleta ARGHIRA **Sergiu Stelian ILIESCU**

GHID PRACTIC DE PROIECTARE ŞI IMPLEMENTARE A APLICAŢIILOR SCADA

EDITURA **CONSPRESS**

2013

Copyright © 2013, Editura Conspress şi autorii

EDITURA CONSPRESS
este recunoscută de
Consiliul Național al Cercetării Ştiințifice din Învățământul Superior

Lucrare elaborată în cadrul proiectului: "*Rețea națională de centre pentru dezvoltarea programelor de studii cu rute flexibile şi a unor instrumente didactice la specializarea de licență şi masterat, din domeniul Ingineria Sistemelor*"

Descrierea CIP a Bibliotecii Naționale a României

HOSSU, DANIELA ; FĂGĂRĂŞAN, IOANA ; DUMITRU, IULIA

Ghid practic de proiectare şi implementare a aplicațiilor SCADA / Daniela Hossu, Ioana Făgărăşan, Iulia Dumitru, ... – Bucureşti : Conspress, 2013
Bibliogr.
ISBN 978-973-100-275-0

I. HOSSU, DANIELA
II. FĂGĂRĂŞAN, IOANA
III. DUMITRU, IULIA
IV. ARGHIRA, NICOLETA
V. ILIESCU, SERGIU STELIAN

681.3

Carte universitară

CONSPRESS
B-dul Lacul Tei nr.124, sector 2,
cod 020396, Bucureşti
Tel.: (021) 242 2719 / 300; Fax: (021) 242 0781

Cuprins

Capitolul 1. Componenta sistemelor SCADA. Arhitectura functionala a sistemelor. Caracteristicile sistemelor. SCADA WinCC 3

1.1. Componenţa sistemelor SCADA _____ 3

1.2. Arhitectura funcţională _____ 6

1.3. Caracteristicile sistemelor SCADA _____ 8

1.4. SCADA WinCC _____ 9

Capitolul 2 Crearea unui proiect. Configurarea conexiunii cu PLC. Configurarea Tag-urilor. 11

2.1. Crearea unui proiect _____ 11

2.2. Selectare Drivere de comunicatie _____ 14

2.3. Configurarea tag-urilor (variabilelor) de proces si grupurilor de tag-uri _ 18

Capitolul 3. Interfata grafica 26

3.1. Configurare aplicatie SCADA _____ 26

3.2. Editorul Grafic _____ 27

Capitolul 4. Afisarea si arhivarea mesajelor 79

4.1. Configurare _____ 79

4.2. Arhivarea _____ 79

4.3. WinCCExplorer / Alarm Logging _____ 80

4.4. Editarea de mesaje bloc _____ 81

4.5. Structura de mesaje pe clase si tipuri _____ 82

4.6. Editarea si configurarea claselor de mesaje_____ 82

4.7. Editarea unui mesaj _____ 84

Capitolul 5. Sisteme SCADA realizate cu automate programabile 93

5.1. Crearea unui nou proiect _____ 96

5.2. Configurarea hardware _____ 97

5.3. Procedura de programare a blocurilor _____ 100

Capitolul 1. Componenta sistemelor SCADA. Arhitectura functionala a sistemelor. Caracteristicile sistemelor. SCADA WinCC

SCADA "*Supervisory Control And Data Acquisition"* se utilizează cu precadere în cazul reţelelor de transport şi distribuţie.

1.1. Componenţa sistemelor SCADA

Sistemele SCADA sunt alcătuite din diferite componente conectate între ele:

Componente de măsurare, acţionare şi automatizare

- *măsurare*: în cazul reţelelor de transport şi distribuţie fluide se măsoară presiunea, temperatura şi debitul, iar pentru reţele electrice se măsoară tensiunea, curentul şi frecvenţa;

  componentele de măsurare pot fi instrumente de măsură cu ieşire digitală sau analogică (conectate la o unitate de conversie analog-digitală). Valoarea digitală a măsurătorii se preia de catre unităţile terminale de prelucrare *(RTU* = remote terminal unit*)*, care evaluează rezultatul măsurătorii, iniţiază efectuarea unor comenzi şi comunică rezultatele măsurătorii către sistemul central de prelucrare.

- *acţionare şi automatizare*: pentru reţele de transport şi/sau distribuţie de fluide: ventile, robinete sau pompe prevăzute cu comandă, etc.; pentru reţele electrice: comutatoare, întrerupătoare, etc.

componentele de acţionare şi automatizare sunt conectate la RTU sau la PLC care pe baza comenzilor sosite de la sistemul central de prelucrare comandă efectuarea unor operaţii.

Componente hardware

- calculatoare, imprimante, monitoare, module inteligente de conducere a proceselor, module de comandă cu logică programată, unităţi de stocare, etc.
- oferă suportul de prelucrare, stocare, introducere, afişare sau imprimare a datelor;
- Din considerente de siguranţă se obişnuieşte folosirea unor elemente redundante pentru a preveni pierderea datelor sau întreruperea funcţionării.

Componente software

- sisteme de operare (de timp real, sau nu), sisteme de achiziţie a datelor, sisteme de gestionare a bazelor de date, programe de simulare, programe de comunicaţii, programe de arhivare/restaurare a datelor. Aceste sisteme oferă suport pentru prelucrare (sisteme de operare, medii de rulare a programelor şi de dezvoltare), asigură mijloace de urmărire, vizualizare şi prelucrare a datelor. Pe baza unor prelucrări anumite componente/module pot iniţia diferite operaţii (comandarea unor elemente de acţionare şi automatizare).
- programele de comunicaţii asigură legăturile între diferite elemente ale sistemului SCADA.
- sistemele de gestiune a bazelor de date (trebuie să existe o bază de date de timp real care înregistreaza valorile momentane şi care asigură

suportul unor prelucrări de timp real; iar pe de altă parte aceste date se vor înregistra pentru analize ulterioare în baze de date convenţionale). Metodele de înregistrare a datelor sunt:

- ***în funcţie de caracteristicile temporale ale înregistrărilor:***
  - continue – monitorizare permanentă a parametrilor procesului;
  - selective – monitorizarea valorilor într-o anumită perioadă;
  - aciclice – determinate de evenimente, de exemplu înregistrarea valorilor curente în cazul condiţiilor critice în care se pot depăşi valorile maxime;
- ***în funcţie de caracteristicile structurale ale înregistrărilor:***
  - de proces – monitorizarea întregului proces;
  - orientată pe subprocese – cu selecţia subproceselor care se monitorizează.

Componente de comunicaţie

- reţele LAN, cablurile reţelelor (cabluri coaxiale, UTP, optice), plăci de reţea
- linii telefonice, modemuri
- mijloace de comunicaţii radio terestre, staţii de emisie-recepţie, relee de transmisie
- mijloace de comunicaţii prin sateliţi, staţii de emisie-recepţie sateliţi

Componente de comunicatie asigură fluxul de date în interiorul sistemului. Pentru a preîntâmpina căderea parţială

sau totală a sistemului se vor folosi mijloace redundante de comunicaţii.

1.2. Arhitectura funcţională

Pentru a oferi suport decizional, sistemele SCADA trebuie să ofere o mare varietate de servicii.

- Serviciul de achiziţie de date, serviciului de comunicare, serviciul de comandă la distanţă.

  Datele culese de modulele de măsurare sunt transmise elementelor locale şi centrale de prelucrare, iar comenzile date de operatori sau procedurile iniţiate de sistemul central de prelucrare sau de modulele locale de decizie sunt transmise elementelor de execuţie. Un serviciu legat de achiziţia de date este verificarea consistenţei datelor, depistarea unor erori de transmisie, determinarea unor eventuale erori de calcul, etc. Dacă a apărut una din condiţiile excepţionale, se declanşează serviciul de alarmare. Mesajele de alarmă se pot afişa pe ecran, pot fi sortate şi filtrate cronologic, după priorităţi, dupa starea curentă a alarmei (activă/inactivă), sursa care a produs mesajul, sau alte opţiuni.

- Interfaţa de operare

  Pentru supravegherea şi conducerea procesului, este necesară urmărirea pe un panou sinoptic a dispunerii reţelei, afişarea

informaţiilor de stare, posibilitatea de iniţiere a unor comenzi, afişarea detaliată a anumitor zone, etc.

- Urmărirea şi analiza tendinţelor de variaţie a variabilelor

  Urmărirea tendinţelor de variaţie a variabilelor de stare este importantă, aceasta putând fi folosită în diagnoză şi în scop predictiv. În acest scop se înregistrează într-o bază de date o istorie a evenimentelor, care pe lângă valorile de stare, conţine şi eventualele alarme şi comenzi date de operatori. Analizele de consum sunt utile pentru depistarea vârfurilor de consum zilnice, săptămânale, lunare şi anuale; pe baza acestor date se pot seta parametrii pentru analizele predictive.

- Serviciul de securitate

  Verificarea accesului în sistem este efectuată de serviciul de securitate, care permite accesul pe bază de parole şi nivele de acces, se pot defini moduri de deconectare a utilizatorilor.

- Serviciul de diagnoză

  Sistemul trebuie să ofere posibilitatea depistării rapide a defecţiunilor din reţea, precum şi a localizării cât mai exacte ale acestora. De asemenea, trebuie să poată oferi toate datele referitoare la posibilele elemente implicate în remedierea defecţiunii.

- Serviciul de arhivare a datelor

  Elaborarea unui sistem de arhivare a datelor este necesară în vederea unor analize / prelucrări ulterioare, precum şi elaborarea unor rapoarte de producţie. În acelaşi timp, datele trebuie să reflecte cât mai exact starea reţelei, iar sistemul să

poată oferi o imagine cât mai completă asupra stărilor, evenimentelor din reţea.

- Serviciul de simulare

  Sistemul de simulare permite analiza unor scenarii de tip "ce se întâmplă, dacă". Acestea se pot referi la impactul unor dezvoltări, extinderi de reţele, situaţii de avarie, etc. Prin simularea reţelei se permite interpolarea valorilor şi în unele puncte în care nu sunt montate instrumente de măsură.

1.3. Caracteristicile sistemelor SCADA

- *Deschiderea*: capacitatea sistemului de a fi supus unei extinderi ulterioare. Aceasta caracteristică se realizează prin utilizarea conceptului de sistem deschis (Open System).

  Deschiderea trebuie să fie prezentă atât din punct de vedere hardware, software (sisteme de operare diferite şi cod portabil), comunicaţii (standarde internaţionale: PROFIBUS şi OPC (OLE for Process Control), MPI, Industrial Ethernet, TCP/IP, etc), cât şi din punct de vedere al administrării datelor (de exp. SQL).

- *Integrabilitatea:* asigură posibilitatea conlucrării cu alte sisteme, cum ar fi de exemplu sistemul informatic al întreprinderii, sistemul de proiectare, sistemul de facturare a consumurilor, staţii de lucru LAN/WAN, sisteme de conducere distribuite, sisteme de conducere a fabricaţiei, sisteme de modelare a proceselor, sisteme de optimizare, etc.

- *Adaptabilitatea*: posibilitatea de a configura componentele conform cerinţelor concrete, chiar în cazul în care aceste cerinţe se modifică pe parcursul duratei de viaţă a sistemului; posibilitatea de a conecta noi echipamente sau programe la sistemul existent.

- *Toleranţă la defect* - se asigură prin redundanţă în module de achiziţie, comunicaţie, unităţile de prelucrare, etc.

- *Disponibilitate ridicată*: se poate realiza prin componenţa modulară şi elemente redundante, precum şi includerea de posibilităţi de autotest, izolare şi ocolire a modulelor defecte.

1.4. SCADA WinCC

In lucrare este folosit ca suport sistemul SCADA WinCC – Siemens.

Functii de baza

- Monitorizarea ecranelor de proces.

Proiectarea ecranului principal prin impartirea in: schema de ansamblu, zona Toolbar, zona de alarme/evenimente, zona de navigare in cadrul ecranelor de proces.
Ierarhizarea ecranelor de proces si navigarea in cadrul ecranelor. Configurarea mesajelor: configurarea ferestrei de mesaje, definirea de clase de mesaje.

Utilizarea de obiecte grafice.

- Monitorizare de alarme, evenimente, arhivare alarme si evenimente. - Achizitia valorilor de masura (functii de arhivare, compresia datelor, limite minime/maxime, etc.).

- Proiectarea de rapoarte

- Comunicatia cu sisteme PLC.

- Interfete standard pentru baze de date si pentru prelucrarea datelor.

- Interfete API .

Functiile complexe se pot realiza utilizand limbajul de programare ANSI-C si/sau Visual Basic Script.

Configuratii

Sistemele distribuite permit distributia tehnologica sau functionala.

Posibilitati de configuratii flexibile ale WinCC**.**

Sistemele Single-user sunt utilizate in aplicatii de mici dimensiuni (numar mic de tag-uri), sau ca interfete operator pentru componente de sistem independente.

Sistem Multi-user permit conectarea pana la 32 de statii de operare la Server.

Server Redundant - cand se opereaza cu doua statii WinCC in paralel, o actualizare a arhivelor (datelor) este facuta automat la runtime sau dupa un defect de Server. Comutarea automata a Clientului la serverul activ garanteaza un proces continuu de executie si operare.

Sisteme distribuite - un Client poate accesa pana la 12 proiecte Server.

Capitolul 2 Crearea unui proiect. Configurarea conexiunii cu PLC. Configurarea Tag-urilor.

2.1. Crearea unui proiect

Pasii pentru crearea unui nou proiect:

In *WinCC Explorer*:

1. Apasa "*File/New*".

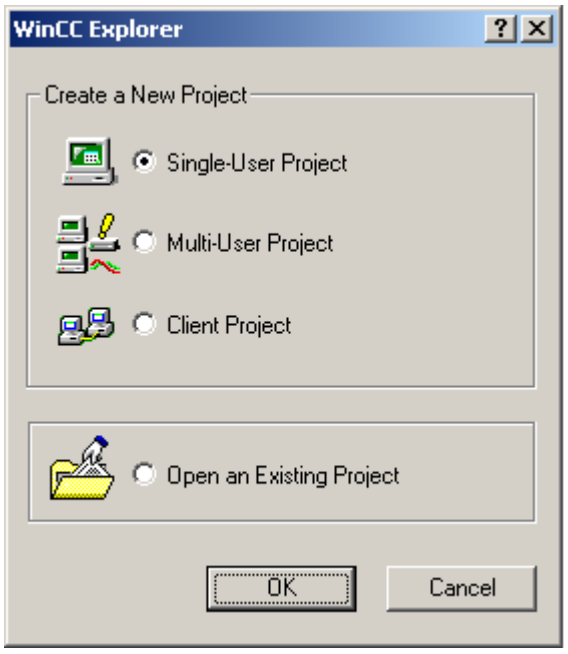


Fig. 2.1

2. Se poate selecta

   - "*Single User*" – un singur utilizator

- „*Multi-User Project*“- Un Server, mai multi Clienti
- „*Client Project*“ – Sisteme distribiute, mai multi Clienti cu propriul lor proiect

3. In fereastra "*Create New Project*" se introduce numele proiectului „*SimClas*“

4. Apasa pe butonul "*Create*".

WinCC creeaza un folder pentru fiecare proiect. Deschiderea unui proiect se face din WinCC Explorer:

Apasa "*File/Open*". Selecteaza *SimClas.MCP*

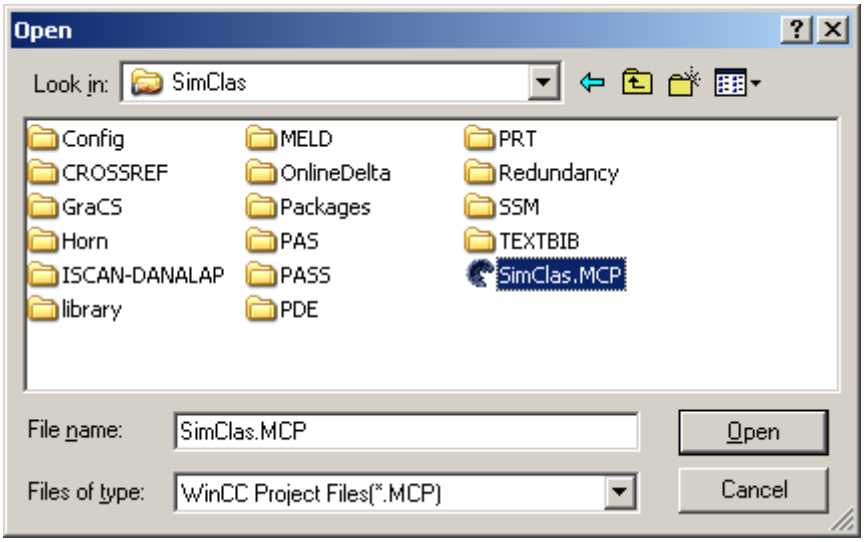


Fig. 2.2

In fereastra *Project properties* se poate selecta *Update Cycles* (aciclic sau 15 optiuni de actualizare de la 250 ms la o ora)

In fereastra *Computer Properties* se pot specifica: *Computer Name*, *Computer Type* (*Server* sau *WinCC Client*), ecranul de start *Start Picture*, etc.

Module functionale:

- Sistem de gestiune a tag-urilor de proces (imagine proces care cuprinde variabile de masura, actionare, referinte, evenimente, arhive), tag-uri interne, definirea de structuri de tag-uri, definirea Conexiunilor.
- Sistem de proiectare a ecranelor, Definire actiuni: C-Programs, VBS
- Sistem mesaje
- Arhivare
- Sistem de rapoarte
- Protectie

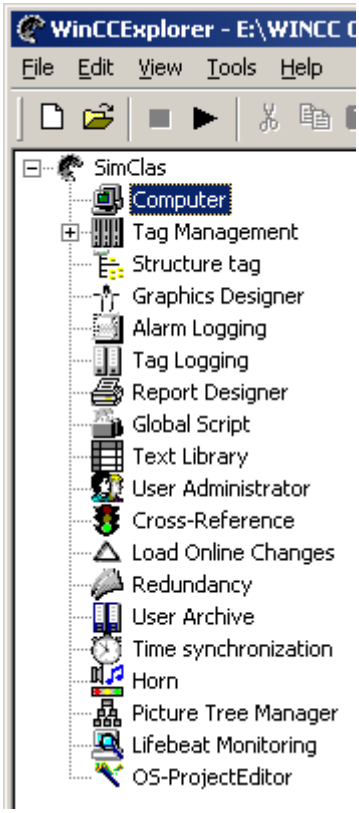


Fig. 2.3

Mecanism de comunicatie

WinCC *Data Manager* se ocupa cu administrarea datelor in sistemul WinCC. Acesta nu este vizibil pentru utilizator. *Data manager* lucreaza cu datele create in proiectul WinCC si stocate in baza de date a proiectului. *Data manager* realizeaza managementul tag-urilor la Runtime.Toate aplicatiile WinCC (*Graphics Runtime, Alarm Logging Runtime* si *Tag Logging Runtime*) cer date de la data manager sub forma de tag-uri WinCC.

Driverele de comunicatie asigura interfata dintre WinCC si proces. Driverul de comunicatie este un C++ -DLL care comunica cu data manager prin interfata specificata de data manager. Aceasta interfata este asa numitul *Channel API*.Tag-urile WinCC sunt incarcate cu date din proces prin acest driver de comunicatie.

Conexiunea hardware la proces este facuta prin *procesorul de comunicatie*. Fiecare canal formeaza interfata cu un sigur driver hardware subordonat deci cu un singur modul de comunicatie.

Driverul de comunicatie WinCC trimite telegrame la PLC prin procesorul de comunicatie; PLC-ul transmite valorile de proces cerute inapoi la WinCC (in telegramele de raspuns).

2.2. Selectare Drivere de comunicatie

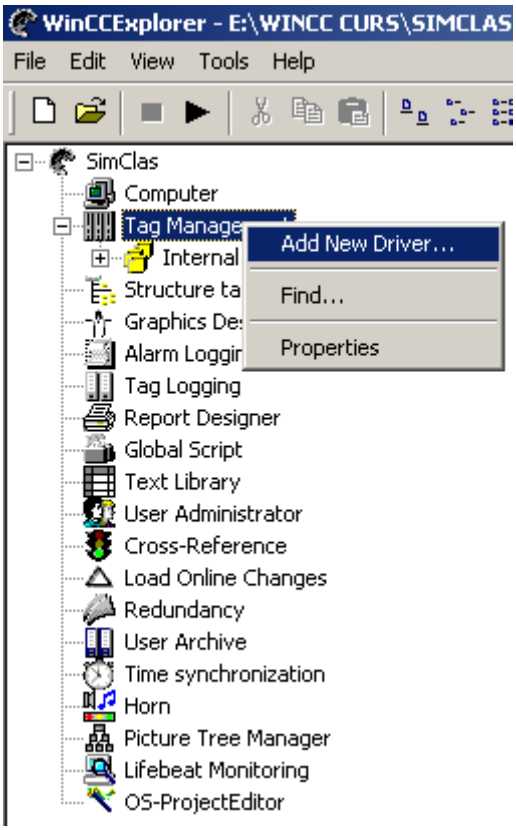

WinCCExplorer - E:\WINCC CURS\SIMCLAS
File Edit View Tools Help
SimClas
Computer
Tag Manage
Internal
Structure ta
Graphics De
Alarm Loggir
Tag Logging
Report Designer
Global Script
Text Library
User Administrator
Cross-Reference
Load Online Changes
Redundancy
User Archive
Time synchronization
Horn
Picture Tree Manager
Lifebeat Monitoring
OS-ProjectEditor
Add New Driver...
Find...
Properties

Fig. 2.4

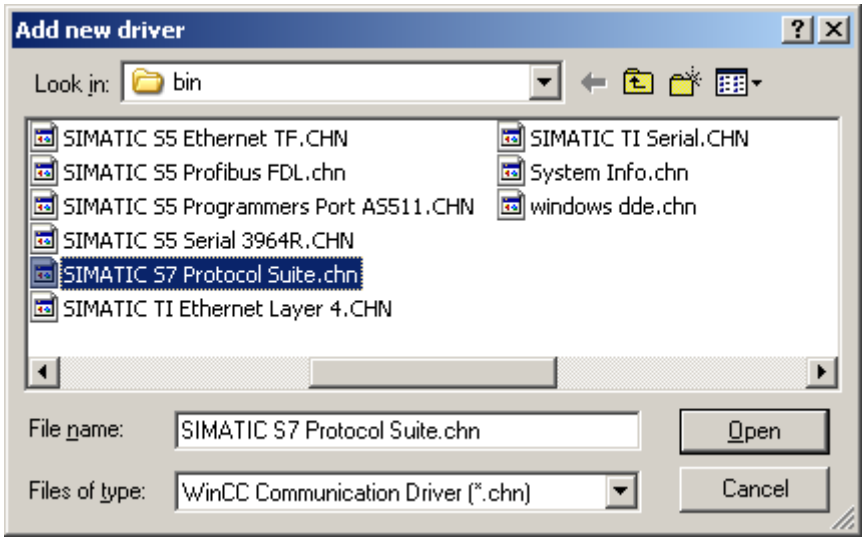

Add new driver
Look in: bin
SIMATIC S5 Ethernet TF.CHN
SIMATIC S5 Profibus FDL.chn
SIMATIC S5 Programmers Port AS511.CHN
SIMATIC S5 Serial 3964R.CHN
SIMATIC S7 Protocol Suite.chn
SIMATIC TI Ethernet Layer 4.CHN
SIMATIC TI Serial.CHN
System Info.chn
windows dde.chn
File name: SIMATIC S7 Protocol Suite.chn
Files of type: WinCC Communication Driver (*.chn)
Open
Cancel

Fig. 2.5

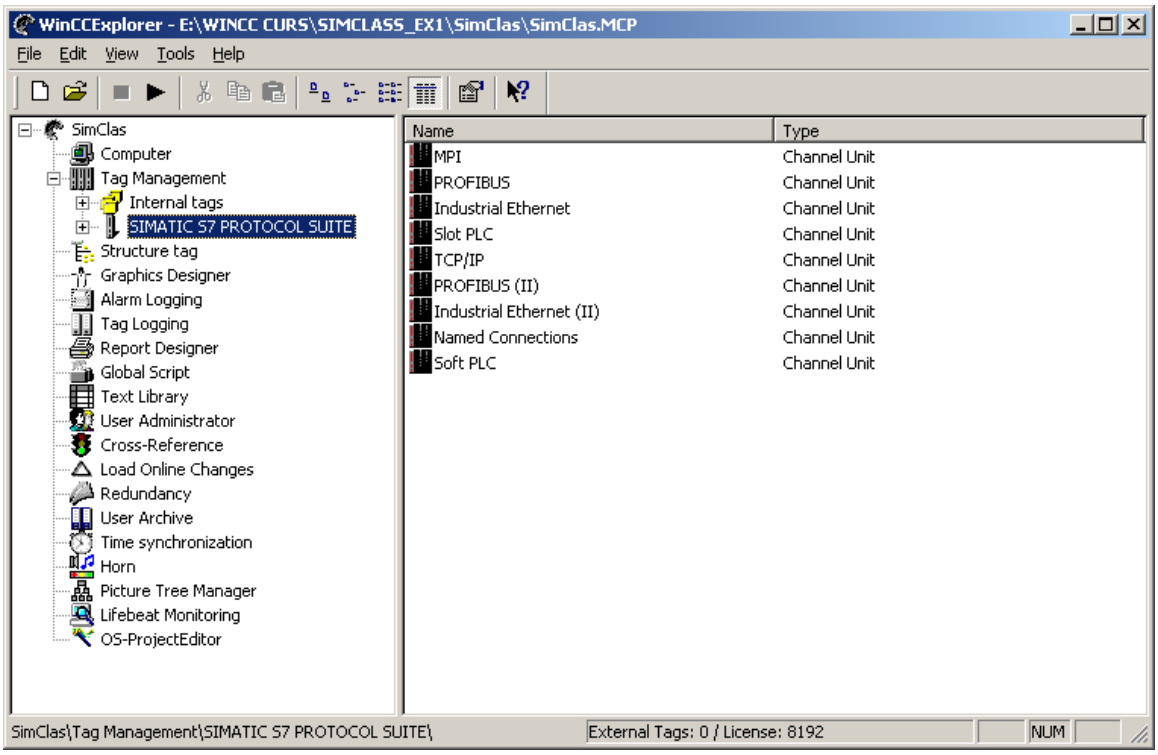


Fig. 2.6

Se selecteaza tipul de conexiune, de exemplu *MPI/New Driver Connection*:

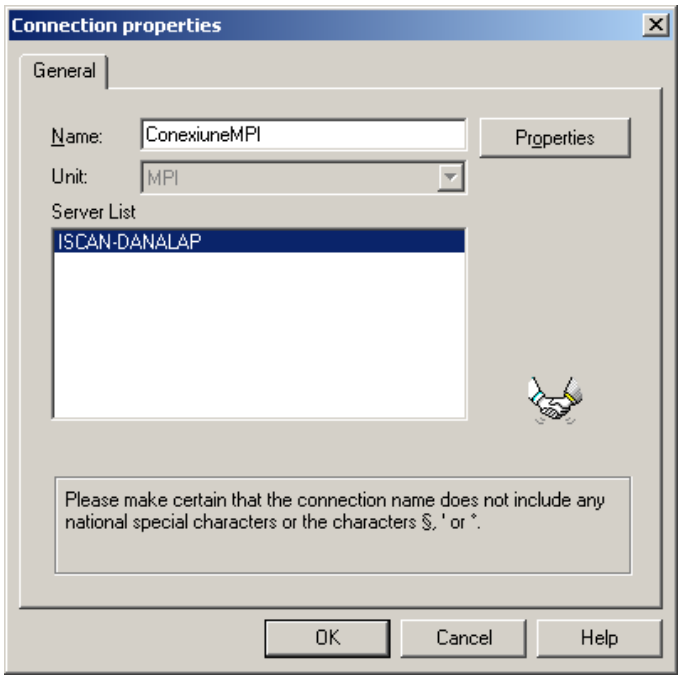


Fig. 2.7

In *Properties* se specifica parametrii conexiunii:

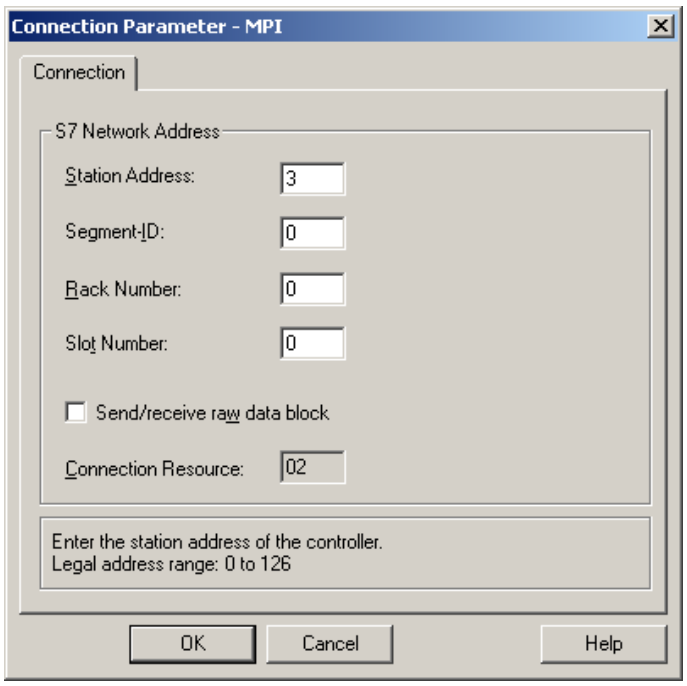


Fig. 2.8

Diagnoza conexiunii se poate face in doua moduri:

1. *Tools /Status of Driver Connections*

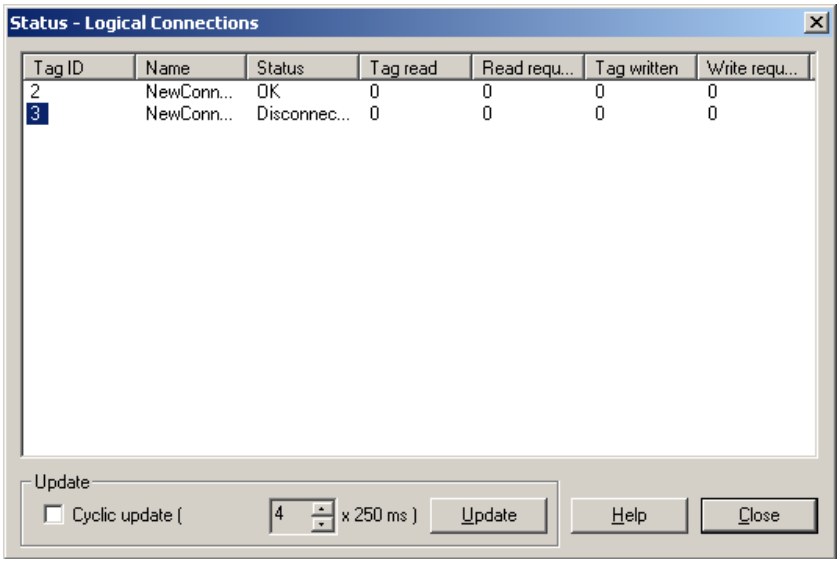


Fig. 2.9

2. *C:\Siemens\WinCC\Diagnose\SIMATIC_S7_PROTOCOL_SUITE_01.LOG*

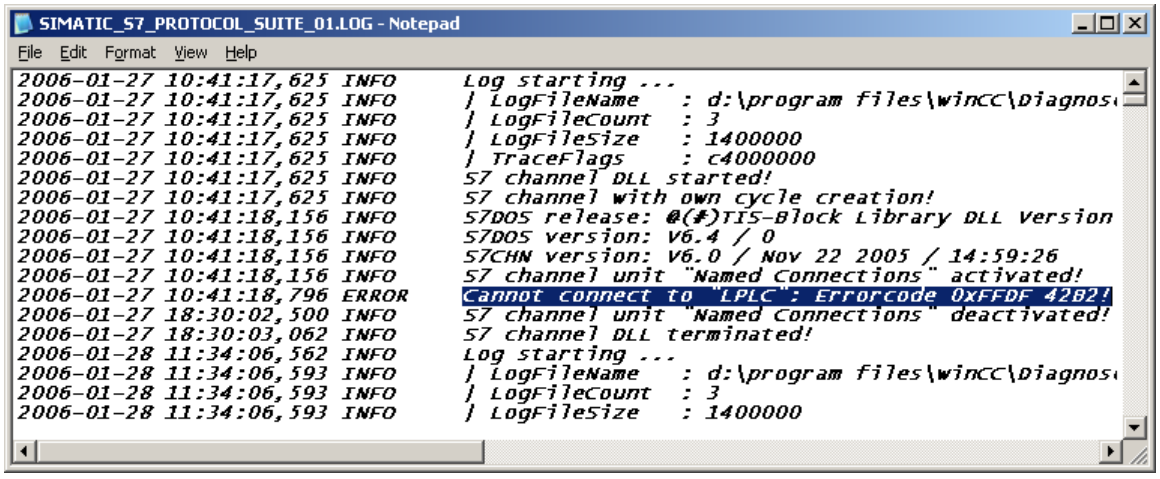

SIMATIC_S7_PROTOCOL_SUITE_01.LOG - Notepad
File Edit Format View Help
2006-01-27 10:41:17,625 INFO Log starting ...
2006-01-27 10:41:17,625 INFO | LogFileName : d:\program files\winCC\Diagnos
2006-01-27 10:41:17,625 INFO | LogFileCount : 3
2006-01-27 10:41:17,625 INFO | LogFileSize : 1400000
2006-01-27 10:41:17,625 INFO | TraceFlags : c4000000
2006-01-27 10:41:17,625 INFO S7 channel DLL started!
2006-01-27 10:41:17,625 INFO S7 channel with own cycle creation!
2006-01-27 10:41:18,156 INFO S7DOS release: @(#)TIS-Block Library DLL Version
2006-01-27 10:41:18,156 INFO S7DOS version: V6.4 / 0
2006-01-27 10:41:18,156 INFO S7CHN version: V6.0 / Nov 22 2005 / 14:59:26
2006-01-27 10:41:18,156 INFO S7 channel unit "Named Connections" activated!
2006-01-27 10:41:18,796 ERROR Cannot connect to "LPLC": Errorcode 0xFFDF 42B2!
2006-01-27 18:30:02,500 INFO S7 channel unit "Named Connections" deactivated!
2006-01-27 18:30:03,062 INFO S7 channel DLL terminated!
2006-01-28 11:34:06,562 INFO Log starting ...
2006-01-28 11:34:06,593 INFO | LogFileName : d:\program files\winCC\Diagnos
2006-01-28 11:34:06,593 INFO | LogFileCount : 3
2006-01-28 11:34:06,593 INFO | LogFileSize : 1400000

Fig. 2.10

In *WinCC Online Help/Error codes with connection failure* sunt listate sunt listate cele mai importante coduri de eroare.

2.3. Configurarea tag-urilor (variabilelor) de proces si grupurilor de tag-uri

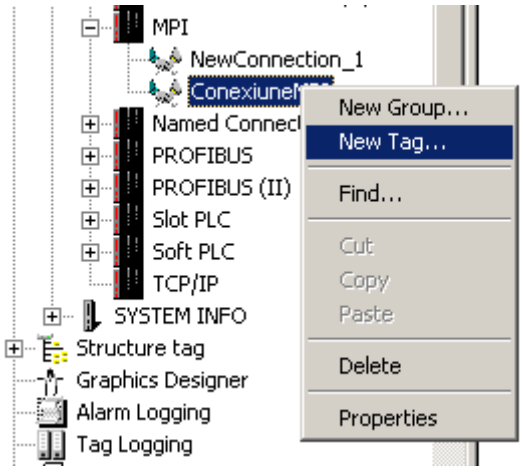


Fig. 2.11

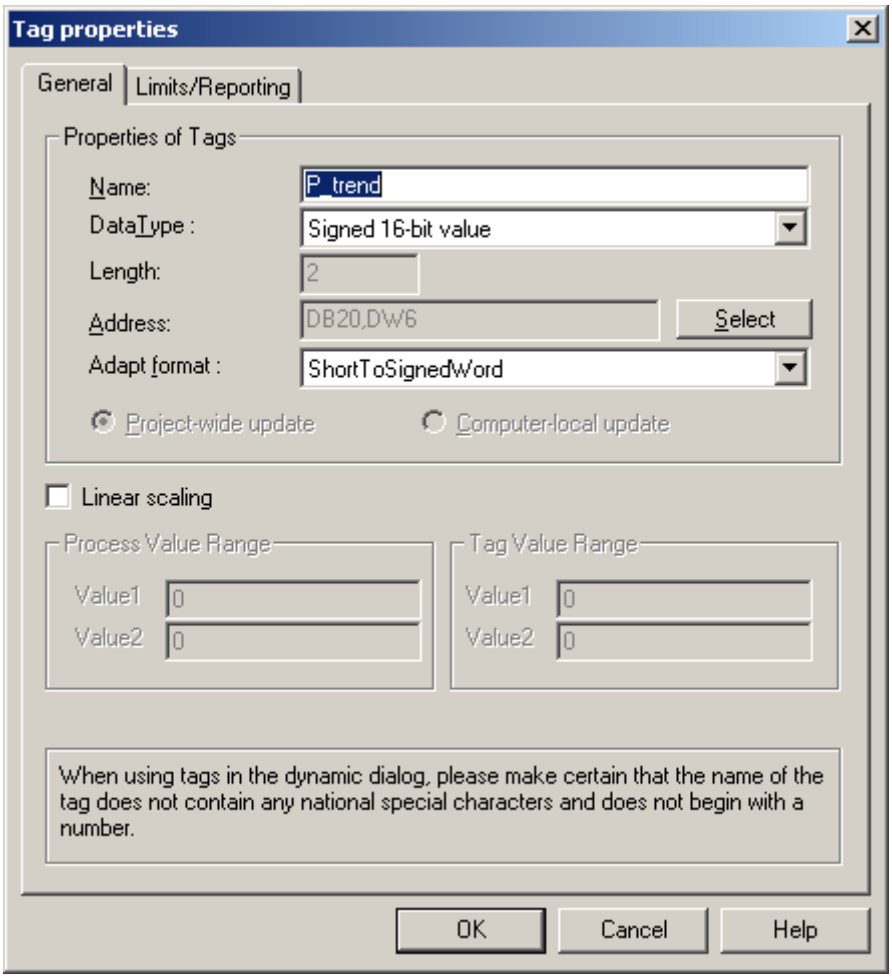


Fig. 2.12

In *Tag properties/Select* se specifica tipul de variabila si adresa.

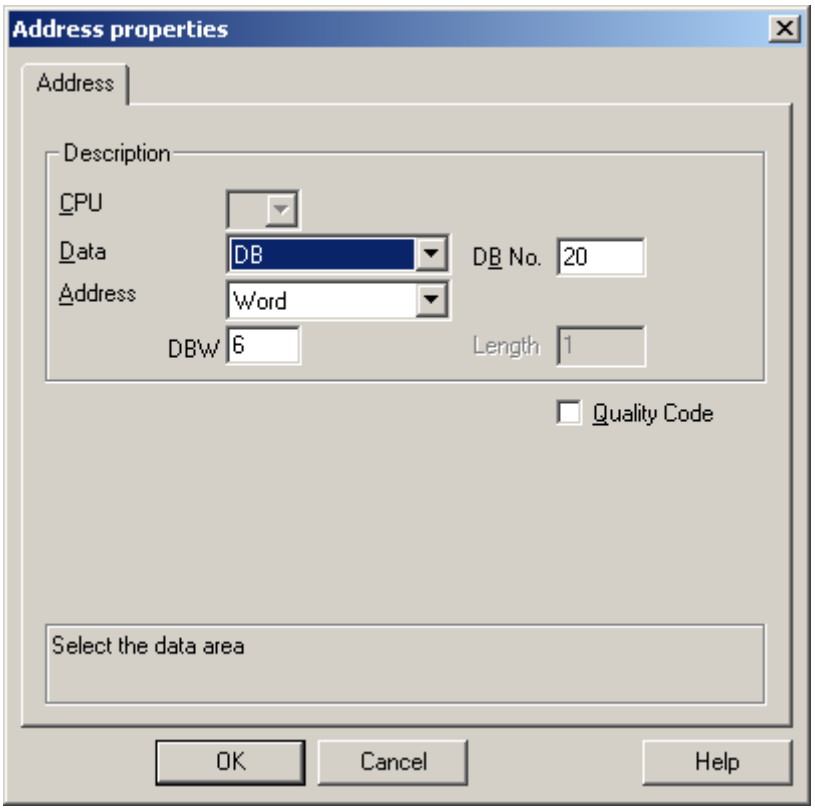


Fig. 2.13

Configurarea Structurilor de tag-urilor (variabile) de proces

In *Structure Tag/New Structure Type* se definesc structurile de tag-uri

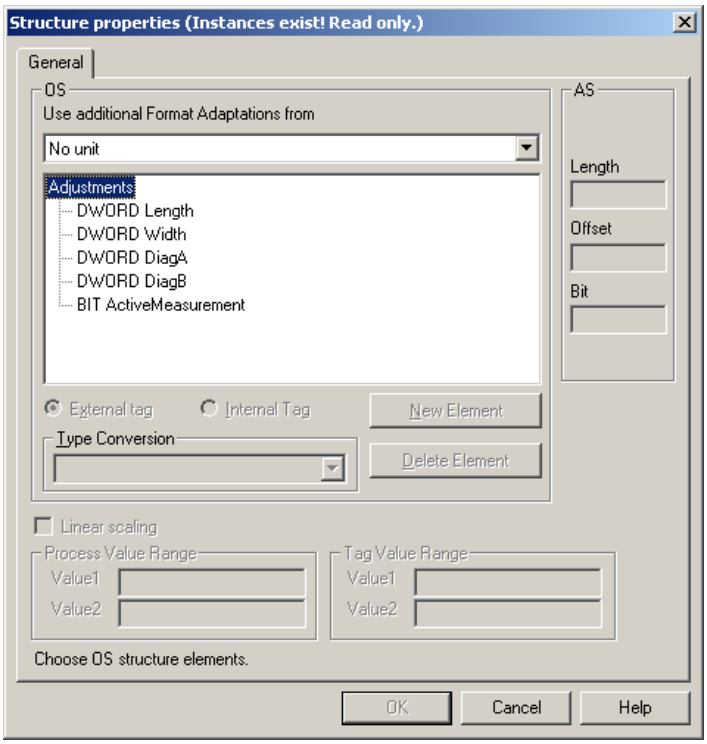


Fig. 2.14

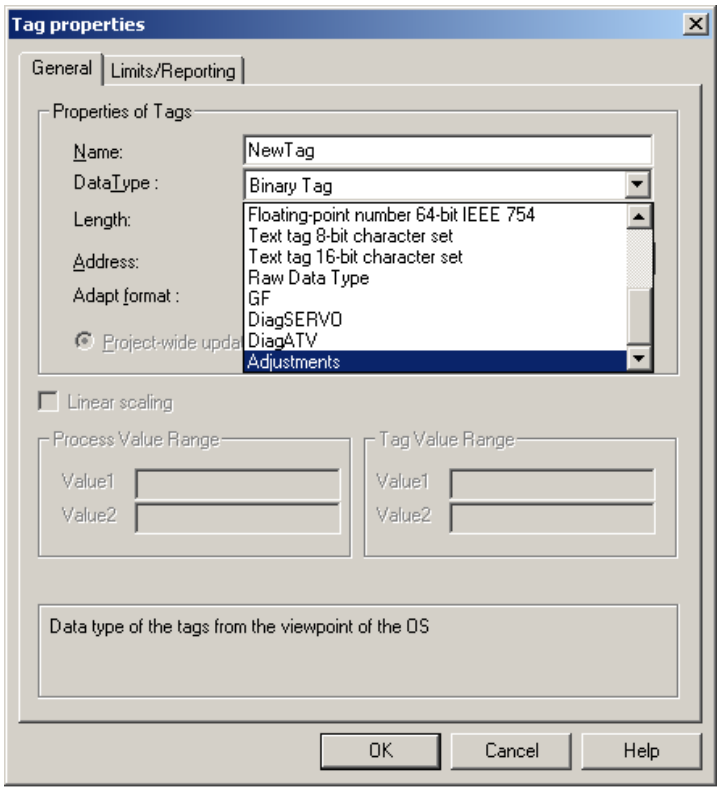


Fig. 2.15

Configurarea tag-urilor (variabilelor) interne si a grupurilor de tag-uri

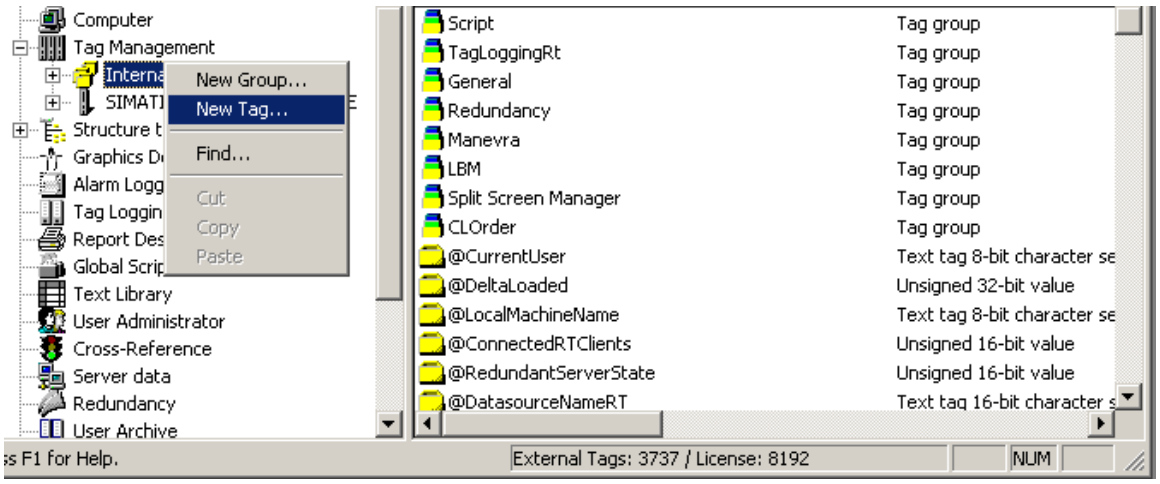


Fig. 2.16

Observatii:

1. Prin apasarea butonului dreapta mouse in *Tag Management/Find* este posibila cautarea conexiunilor, tag-urilor sau grupurilor de tag-uri.

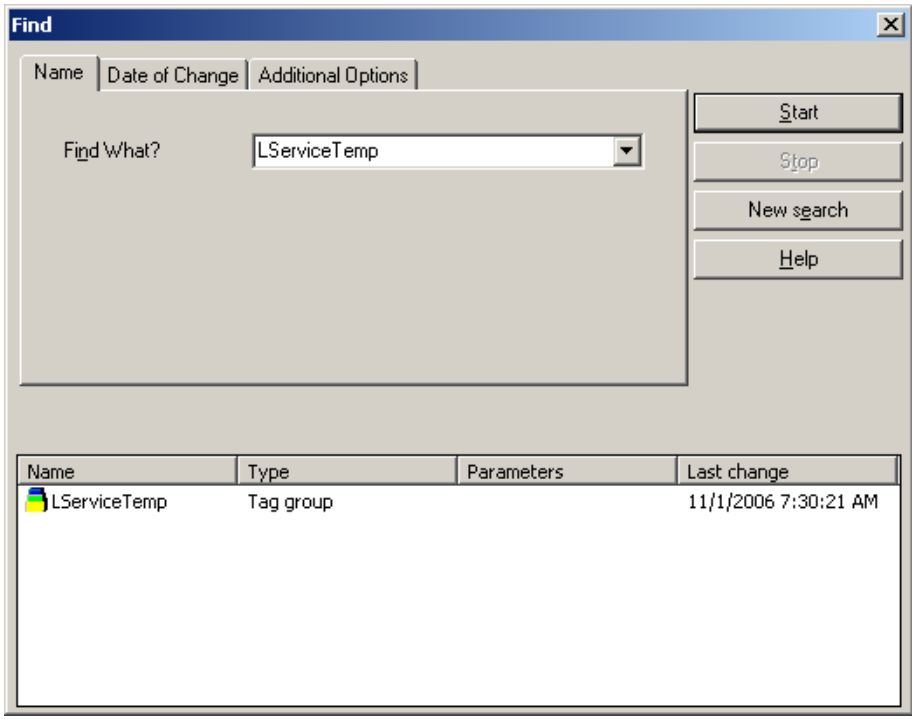


Fig. 2.17

In *WinCCExplorer/Cross-Reference* se pot vizualiza tag-urile (si cele create prin *VBS Actions*), se pot face filtrari ale bazei de date, se pot face foarte simplu operatii de find/replace.

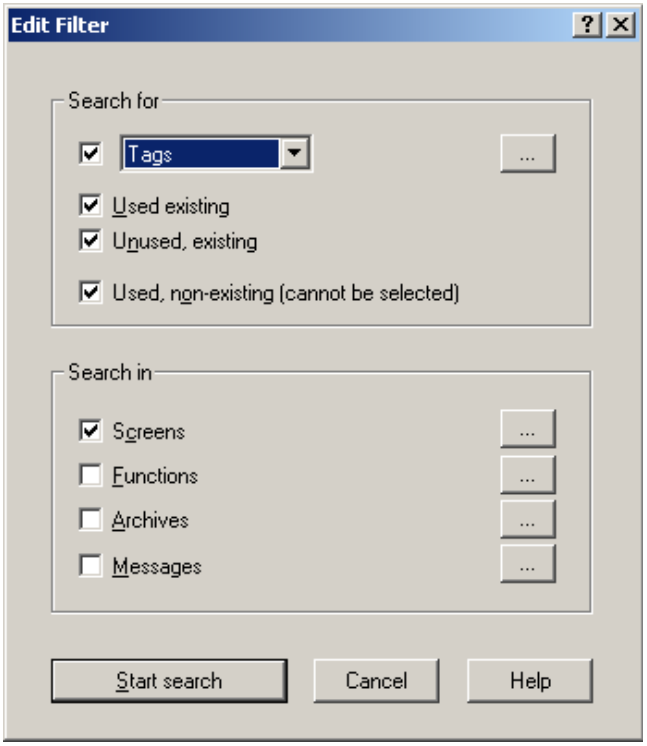


Fig. 2.18

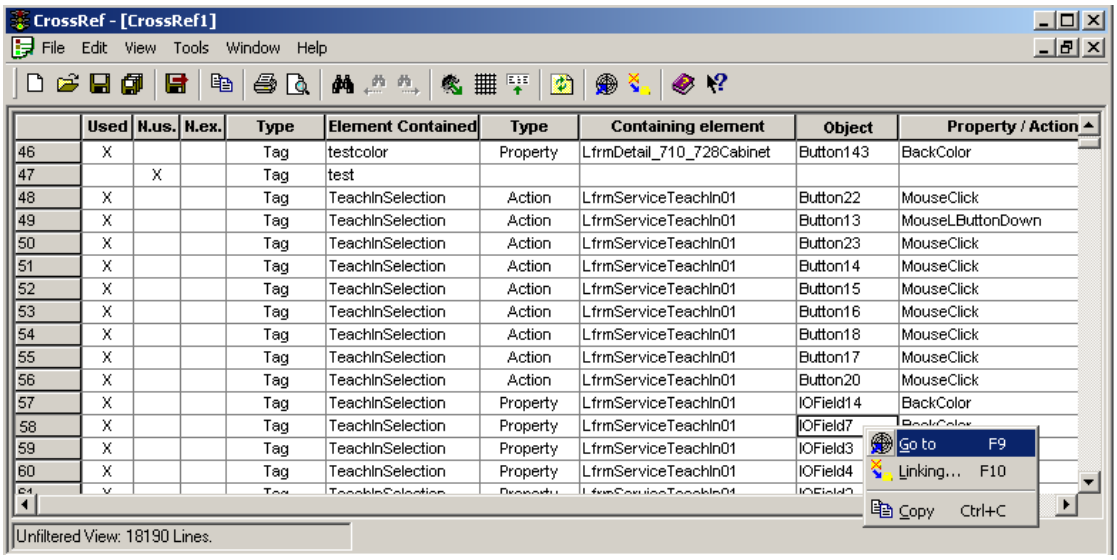

CrossRef - [CrossRef1]
File Edit View Tools Window Help
Used N.us. N.ex. Type Element Contained Type Containing element Object Property / Action
46 X Tag testcolor Property LfrmDetail_710_728Cabinet Button143 BackColor
47 X Tag test
48 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button22 MouseClick
49 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button13 MouseLButtonDown
50 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button23 MouseClick
51 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button14 MouseClick
52 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button15 MouseClick
53 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button16 MouseClick
54 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button18 MouseClick
55 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button17 MouseClick
56 X Tag TeachInSelection Action LfrmServiceTeachIn01 Button20 MouseClick
57 X Tag TeachInSelection Property LfrmServiceTeachIn01 IOField14 BackColor
58 X Tag TeachInSelection Property LfrmServiceTeachIn01 IOField7
59 X Tag TeachInSelection Property LfrmServiceTeachIn01 IOField3
60 X Tag TeachInSelection Property LfrmServiceTeachIn01 IOField4
Go to F9
Linking... F10
Copy Ctrl+C
Unfiltered View: 18190 Lines.

Fig. 2.19

In *WinCC/Tools/WinCC Tag Simulator* se pot realiza simulari ale tag-urilor (tag-urile de proces pot fi simulate numai daca s-a realizat conexiunea cu PLC-ul).

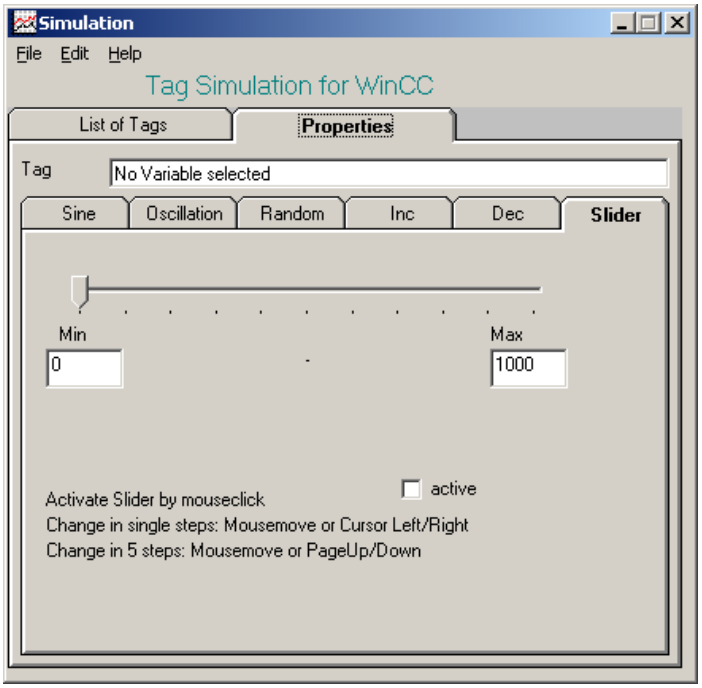


Fig. 2.20

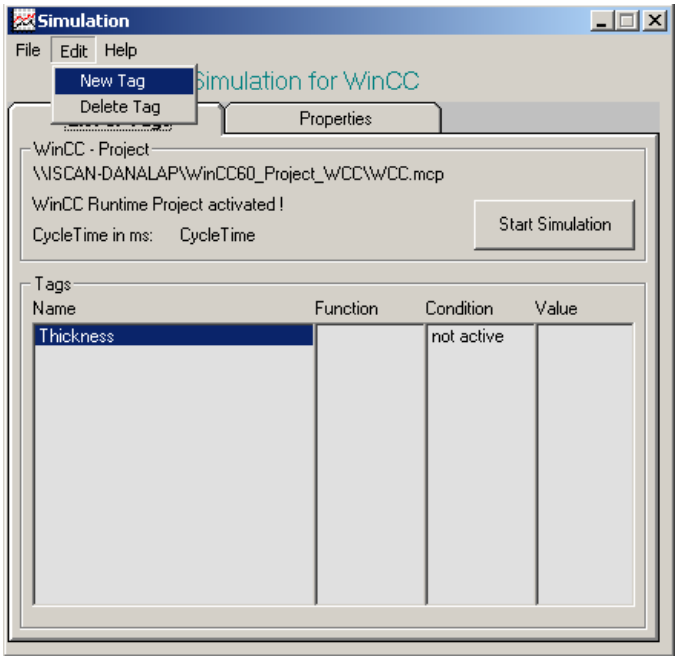


Fig. 2.21

Exercitiu:

- Crearea unui proiect,

- Configurarea conexiunii cu PLC – *Simatic S7 Protocol Suite, MPI*

- Configurarea de Tag-uri de proces si grupuri de Tag-uri,

Grup de tag-uri:

CapDataTest

Tag-uri:

ConStart, Unsigned 16 bit value, DB329 DW0
CD_Picking, Unsigned 8 bit value, DB320 DBB16
CD_PlateID, Unsigned 32 bit value, DB320, DD2
CD_Length, Unsigned 32 bit value, DB320, DD8
CD_Width, Unsigned 16 bit value, DB320 DW14

Lstate

Tag-uri:

LStatGeneral, Unsigned 16 bit value, DB1002 DW1158
LStatSensor, Unsigned 16 bit value, DB1002 DW266
LstatMotor, Unsigned 32 bit value, DB1002, DD302

- Configurarea de Tag-uri interne,

- Tag-uri interne

Grup de tag-uri:

General

Tag-uri:

CmdAcceptNewCoef, Binary Tag

CmdRemove, Binary Tag

Capitolul 3. Interfata grafica

3.1. Configurare aplicatie SCADA

Conectarea la proces :

- se specifica tipul de PLC utilizat pentru schimbul de date.
- se specifica nivelul fizic pentru transferul datelor.
- se realizeaza o dimensionare a procesului, se specifica numarul si tipul de I/O.

Monitorizarea:

- se imparte procesul in ecrane de proces, se defineste un ecran de start, se defineste modul de selectie pentru ecranele de proces. Ecranele de proces reprezinta sistemul sau componente ale sistemului si permit operarea/supravegherea sistemelor/componentelor in regim automat/manual. Ecranele constau din componente grafice statice si dinamice. Obiectele care nu se schimba (in forma sau culoare) sunt componente statice. Toate obiectele care afiseaza o valoare de proces sau stare de proces sunt componente dinamice.
- se specifica componentele (partile) statice ale imaginii (text, linie, cerc, dreptunghi, culoare, etc.) si partile active ale imaginii

(valori de intrare/iesire, operatii de comutare, curbe de variatie, mesaje, etc.)

- se specifica tag-urile pentru dinamica
- se defineste zona de ansamblu care contine informatii generale (mesaje), zona de lucru in care se monitorizeaza evenimente din proces si zona de butoane (cum ar fi butoanele utilizate pentru selectarea ecranelor de proces).

Achizitia valorilor de masura si arhivarea mesajelor

- Se defineste tipul de afisarea a valorilor de masura (tabele, etc.)
- Se definesc valorile de masura care trebuie arhivate.
- Se definesc mesajele (cu sau fara confirmare / arhivare, etc.)

Rapoarte

Parole

- (definire) grup
- (definire) utilizator

3.2. Editorul Grafic

WinCC Explorer/Graphics Designer se utilizeaza pentru dezvoltarea ecranelor de proces cu ajutorul componentelor statice si dinamice.

Paleta de obiecte grafice

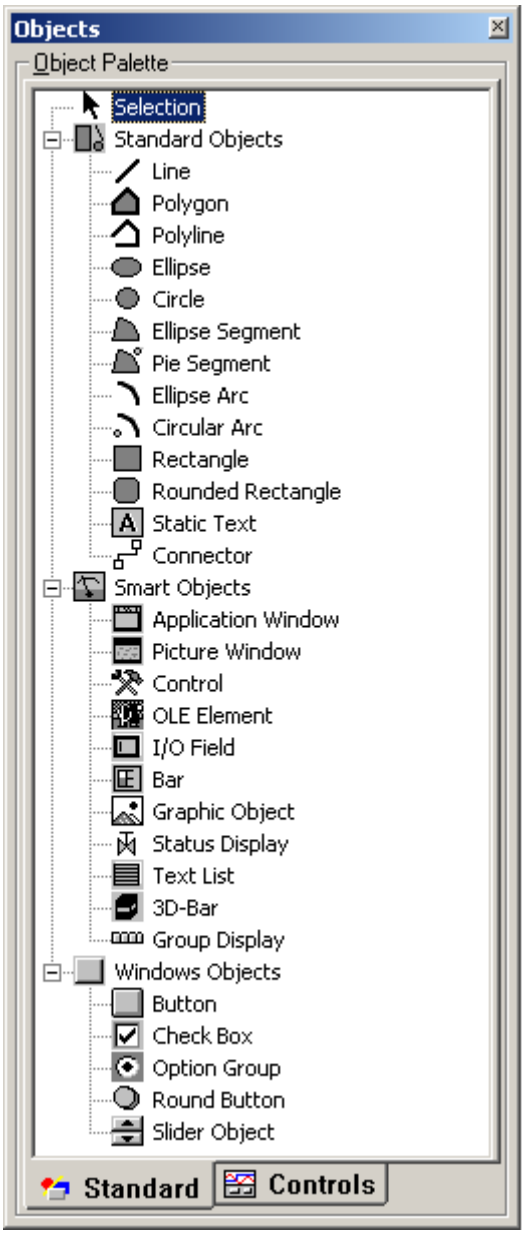

Objects
Object Palette
Selection
Standard Objects
Line
Polygon
Polyline
Ellipse
Circle
Ellipse Segment
Pie Segment
Ellipse Arc
Circular Arc
Rectangle
Rounded Rectangle
Static Text
Connector
Smart Objects
Application Window
Picture Window
Control
OLE Element
I/O Field
Bar
Graphic Object
Status Display
Text List
3D-Bar
Group Display
Windows Objects
Button
Check Box
Option Group
Round Button
Slider Object
Standard
Controls

Fig. 3.1

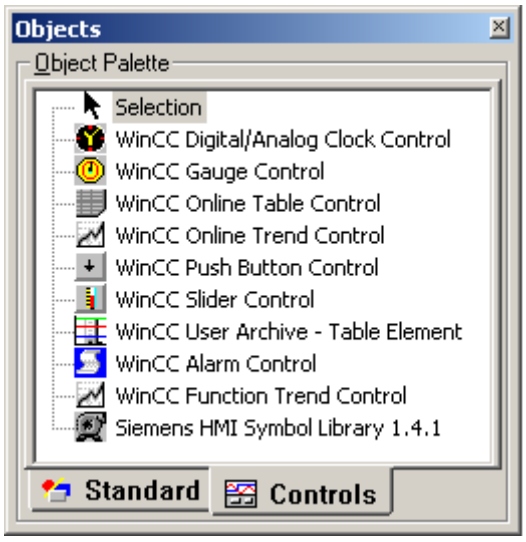


Fig. 3.2

Cu butonul stanga mouse se selecteaza un obiect din paleta de obiecte si se plaseaza in interfata de configurare.

Standard Objects

Obiectele standard sunt compuse din obiecte grafice si text static. Conectorul este un obiect linie ale carui capete trebuie conectate cu punctele de conectare ale altor obiecte.

Smart Objects

Application windows sunt obiecte alimentate de sistemul de rapoarte, sistemul de diagnoza sau de aplicatii externe. Proprietatile externe (pozitie, dimensiune, etc) sunt configurate in Graphics Designer si sunt transferate aplicatiilor externe la runtime.

Picture windows sunt obiecte configurate prin pozitie, dimensiuni si alte propriatati si care pot fi facute dinamice. O aplicatie importanta este deschiderea unui alt ecran care se deschide in "*picture window*" (continutul se modifica dinamic la runtime prin "*picture name*").

Controls sunt obiecte utilizate pentru a implementa elemente Windows. Un control OLE are atribute afisate in fereastra "Object Properties".

OLE elements pot fi integrate in ecranele grafice (obiecte grafice, obiecte audio sau video, Excel chart, document word, etc).

I/O Fields reprezinta campuri de intrare/iesire.

Bar (Graphs) afiseaza valori din proces in forma analogica.

Graphic Objects sunt utilizate pentru a integra obiecte grafice externe (.wmf, .emf, .bmp, .gif, .jpg, .jpeg, .dib) intr-un ecran din Graphics Designer. Status Display este utilizat pentru afisarea schimbarilor dinamice ale obiectelor grafice (.wmf, .emf, .bmp, .gif, .jpg, .jpeg, .dib).

Windows Objects

Buttons sunt utilizate pentru a opera evenimente din proces. Check Boxes sunt necesare pentru selectia mai multor operatii. Option Groups sunt similare cu check box, insa este permisa executia unei singure operatii.
Round Buttons sunt utilizate pentru a opera evenimente din proces. Sliders sunt utilizate pentru ajustarea valorilor analogice din proces.

Pentru a activa modul *Runtime*, trebuie definit un ecran de start. Acest ecran este baza pentru intreaga structura de ecrane care se vor configura pentru operare.

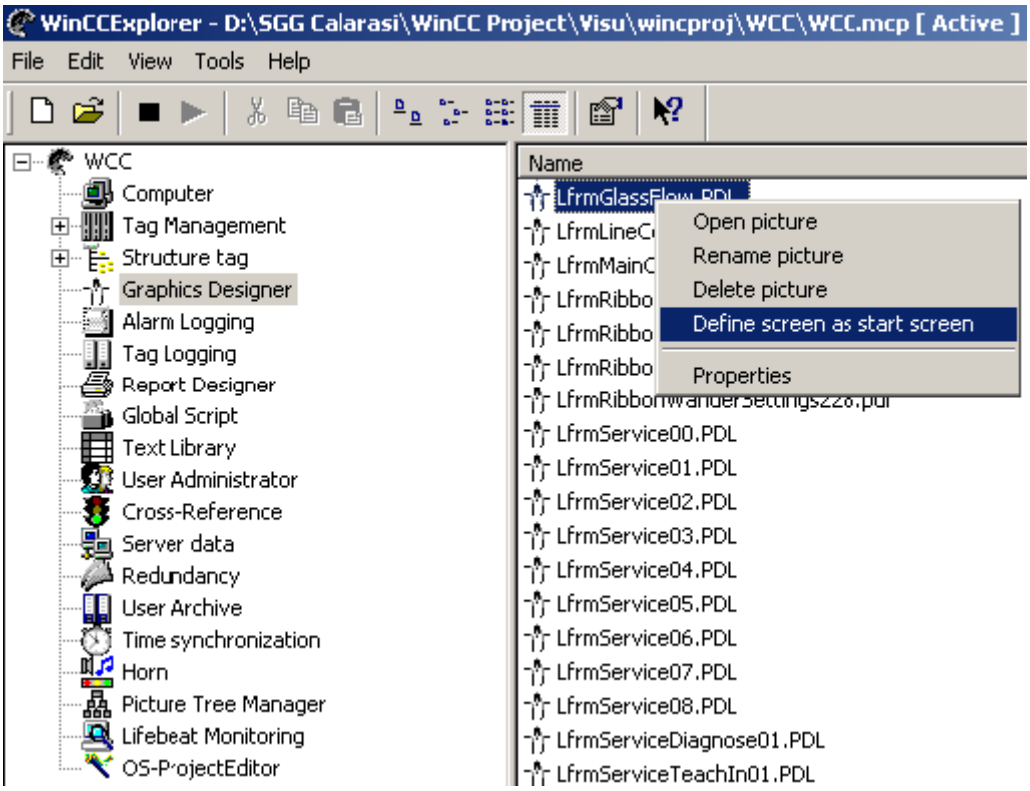


Fig. 3.3

Activare Runtime

1. WinCC Explorer.

2. *File / Activate*

Iesire din WinCC Explorer

1. WinCC Explorer.

2. *File / Exit*.

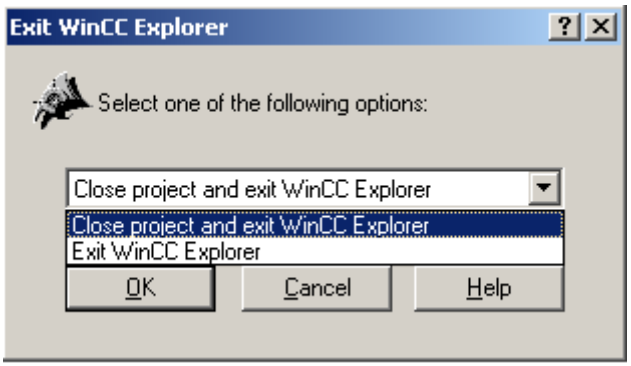


Fig. 3.4

In scopul optimizarii selectiei imaginilor si a comunicatiei se recomanda utilizarea unui *default trigger* pentru toate obiectele imaginii (se recomanda uzual o valoare de 2 secunde).

1. WinCC Explorer.

2. *Tools*

3. *Setings/Default Object Settings* .

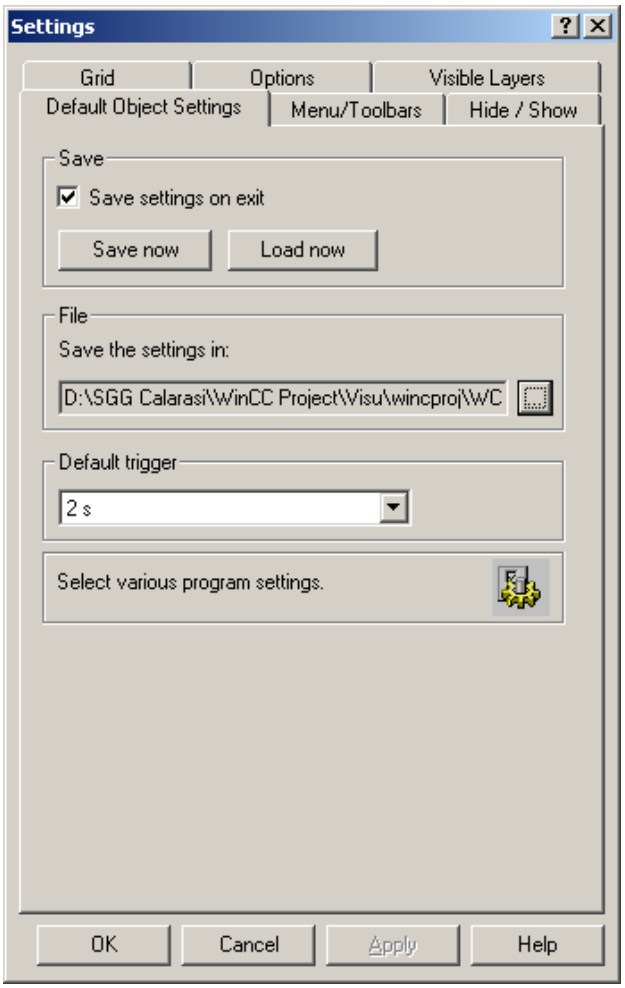


Fig. 3.5

Proiectarea imaginilor de proces dinamice

Configuration Dialog

Dynamic Wizard

Tag Connection

Dynamic Dialog

Direct Connection

Property actions (ANSI C)

Event actions (ANSI C)

VBS Visual Basic Script Properties

VBS Events

Configuration Dialog este o metoda de a configura aplicatii standard (obiecte) cum ar fi *IOField*. Nu toate obiectele au un asemenea dialog. Configuration Dialog se poate activa prin *Tools / Settings / tab: Options*.

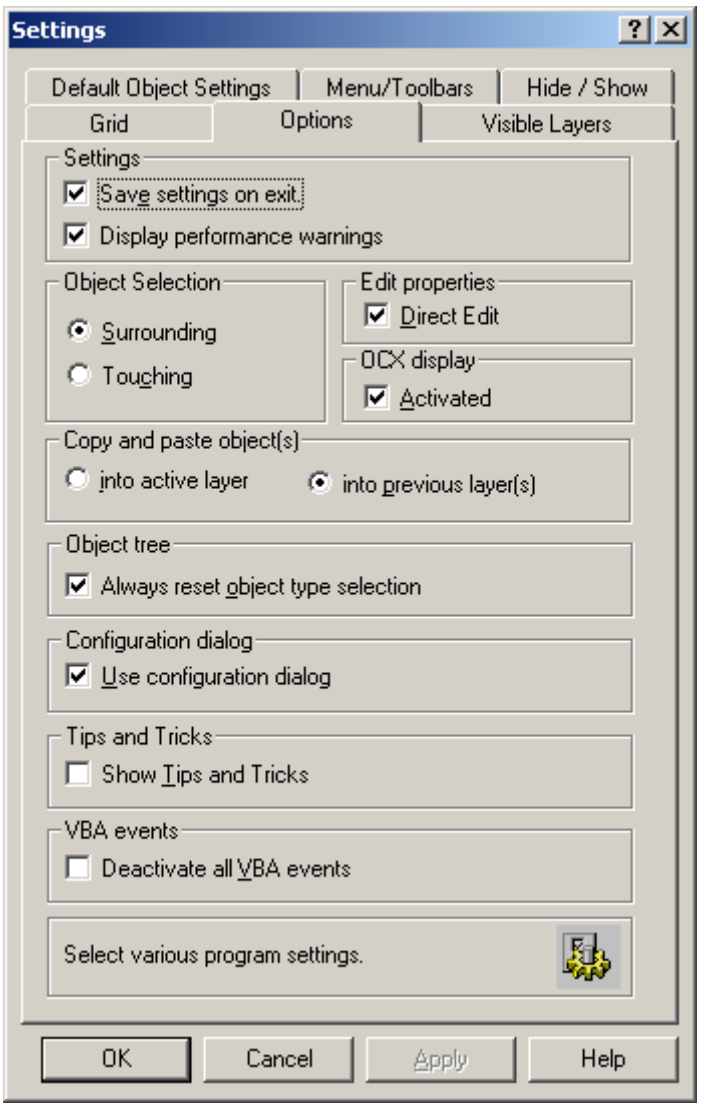


Fig. 3.6

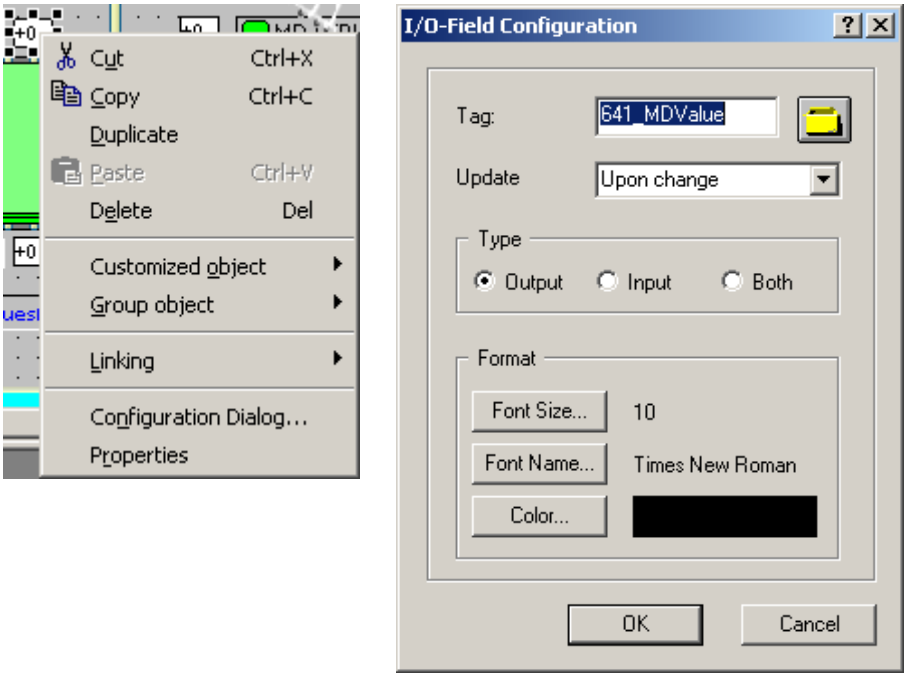


Fig. 3.7

Dynamic Wizard permite utilizarea si configurarea obiectelor complexe (bar, button, etc.). Dynamic Wizard genereaza un C-script. Dynamic Wizard se poate activa prin *View /Toolbars*.

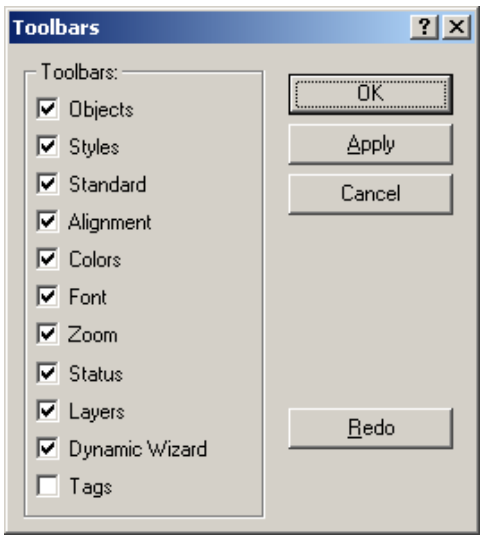


Fig. 3.8

O aplicatie uzuala este configurarea unui *buton* de iesire din proiect sau runtime (se selecteaza obiectul (*buton*), dublu click pe *Dynamic Wizard / Exit WinCC Runtime*).

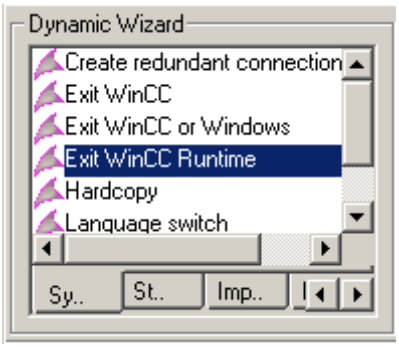


Fig. 3.9

Tag Connection permite conectarea directa a tagurilor interne si de proces.

Selecteaza un obiect din ecran, *Object Properties/ Properties* / alege *proprietate* (*Line Background*) / alege *Tag* (in casuta de dialog alege tag-ul dorit).

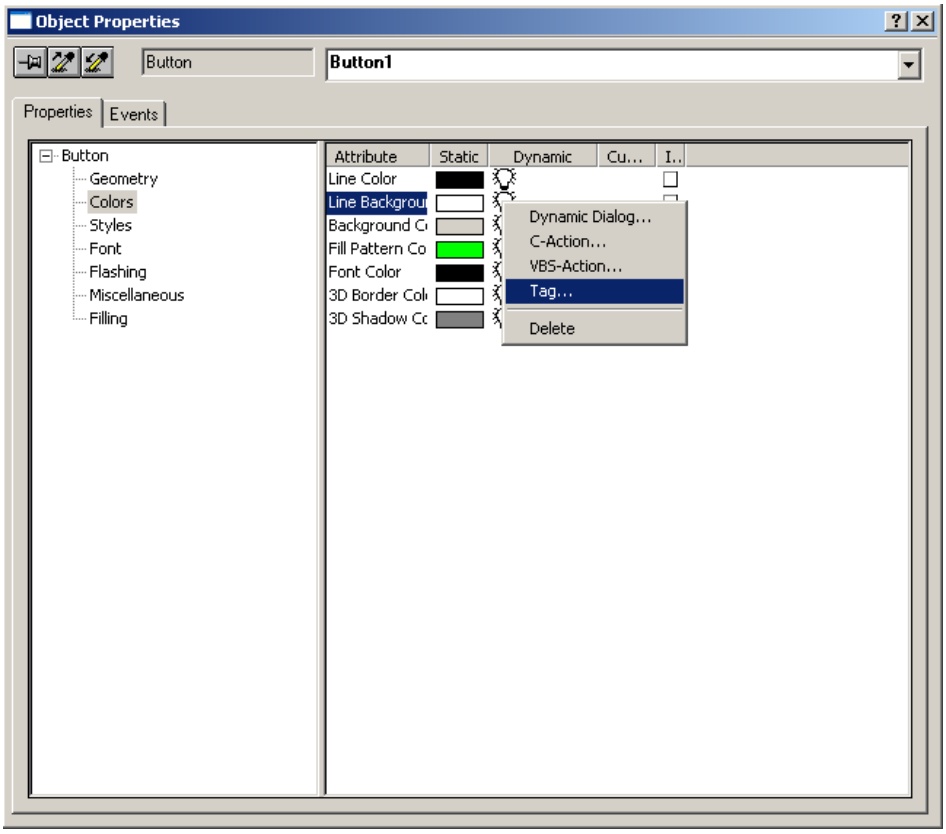


Fig. 3.10

Dynamic Dialog permite configurarea obiectelor complexe obtinand performante ridicate la Runtime (schimbarea culorii pentru diferite intervale sau limite – se introduc limitele si parametrii care trebuie schimbati).

Selecteaza un obiect din ecran / *Object Properties / Properties / Dynamic Dialog*.

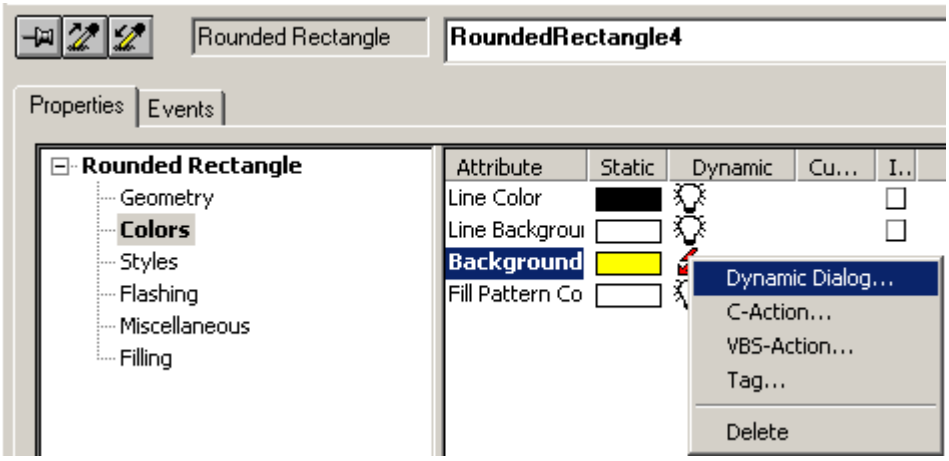


Fig. 3.11

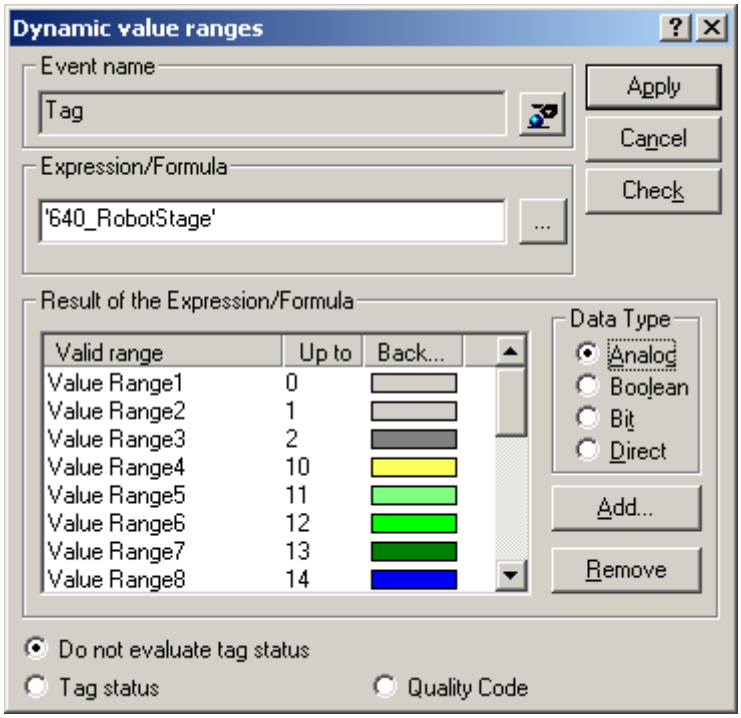


Fig. 3.12

Cu ajutorul functiei ***Dynamic Dialog*** este posibila evaluarea starii unui tag de proces (buton *Emergency*: *ON /OFF*).

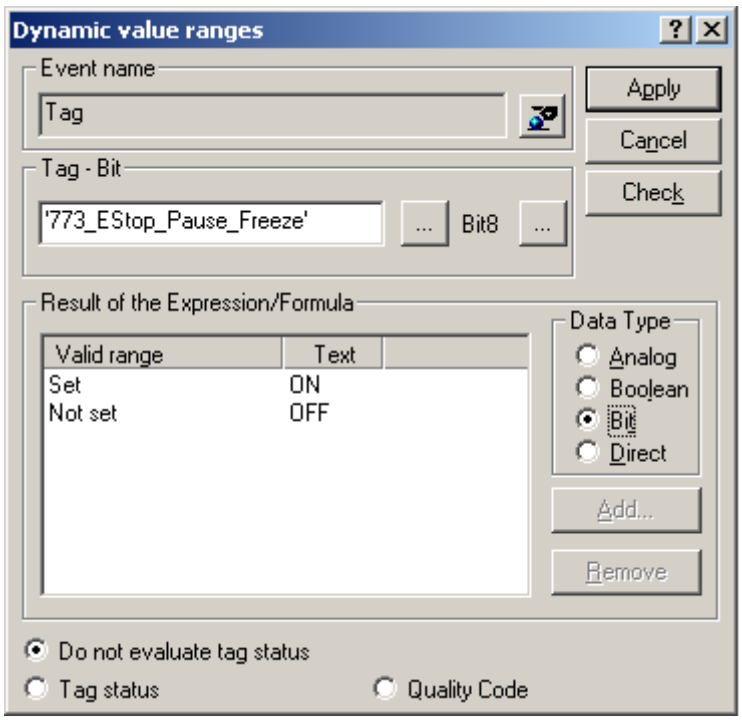


Fig. 3.13

Direct Connection permite conectarea directa a proprietatilor obiectelor la taguri obtinand performante ridicate la Runtime (de exemplu conectarea directa a unui tag de tip *string* la proprietatea *text* a unui obiect *static text*.

Selecteaza un obiect din ecran / *Object Properties* / *Events* / *Action* / *Direct Connection*.

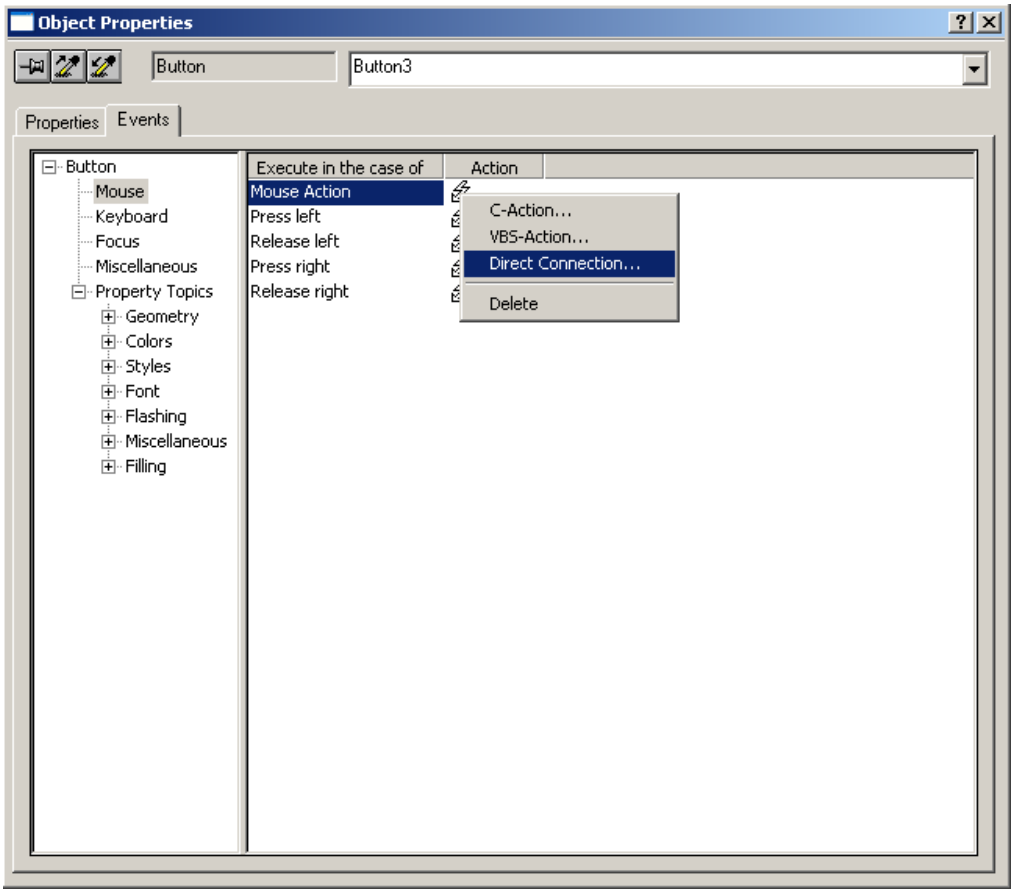


Fig. 3.14

Direct connection permite conectarea catorva proprietati si actiuni intre ele, fara a fi necesara formularea de actiuni in C.

Exemplul 1.

Setarea unei referinte.

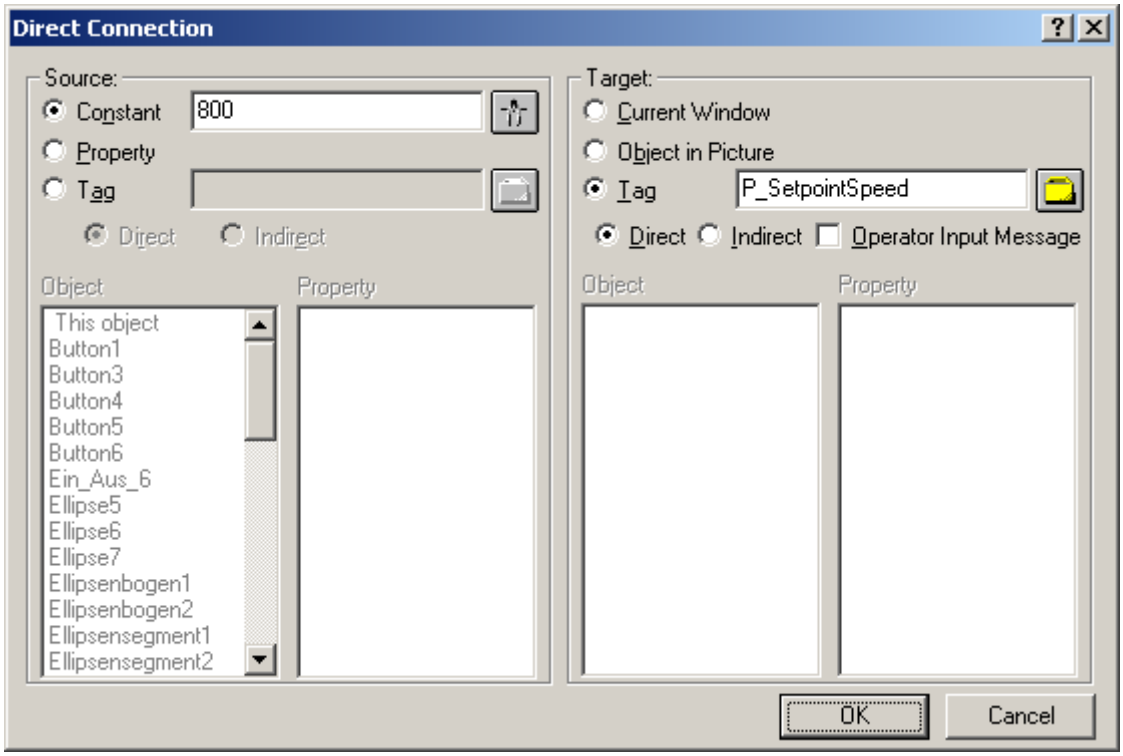


Fig. 3.15

Exemplul 2.

Navigarea intre ecrane - intoarcerea la ecranul de start.

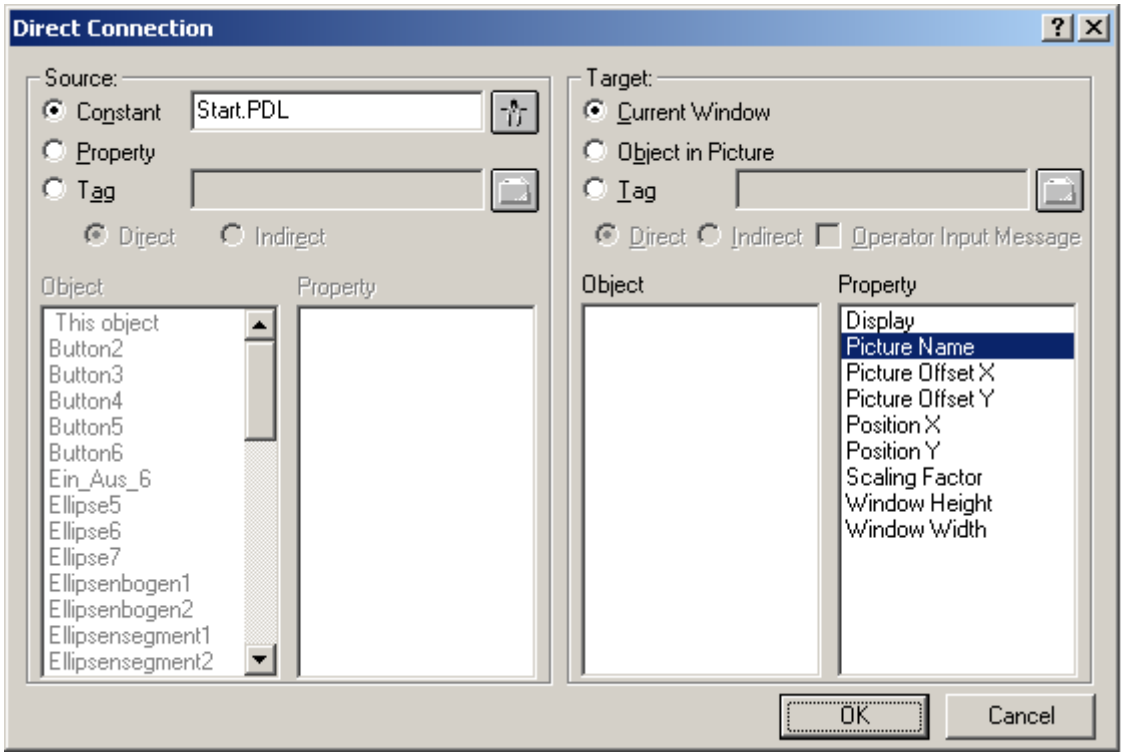


Fig. 3.16

Property Actions Numeroase proprietati ale obiectelor pot fi configurate dinamic, (*color*, *size*, *position*, *font size*, etc, prin intermediul tag-urilor de proces sau a tag-urilor interne). Se pot configura actiuni in C si VBS. Actiunea se realizeaza printr-un timer sau schimbare de tag.

Exemplul 1.

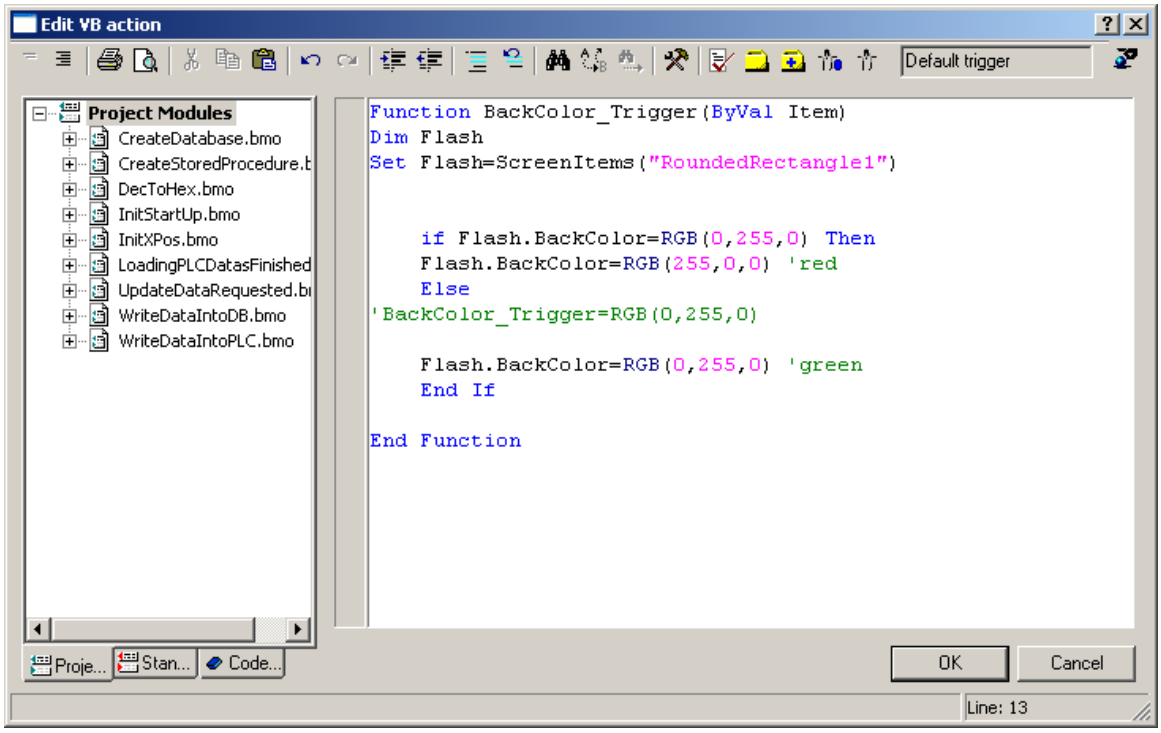


Fig. 3.17

Culoarea alterneaza la interval de 1s in rosu si verde (flash). Functia este realizata cu valoarea returnata sau cu proprietatea "*BackColor*".

Event Actions Se poate utiliza C si VBS pentru configurarea unor actiuni ca urmare a realizarii unui eveniment (*mouse click*).

Selecteaza un obiect din ecran / *Object Properties* / alege *Properties* sau *Events* / *Action* / selecteaza *C-Action* sau *VBS-Action*.

Exemple *VBS-Action*

Exemplul 1.

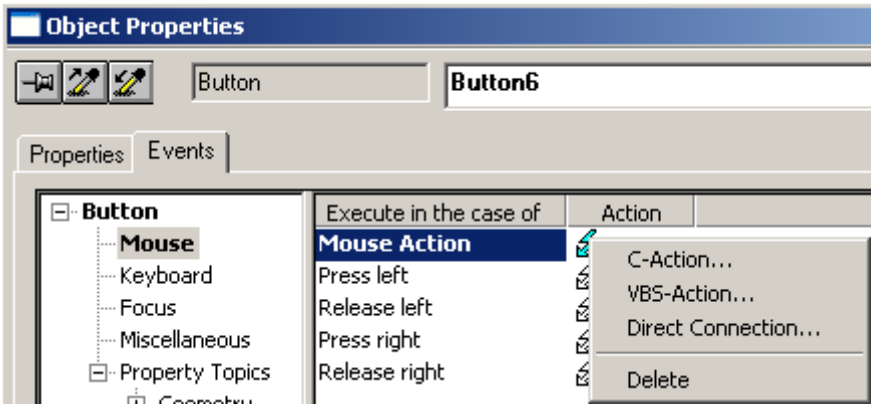


Fig. 3.18

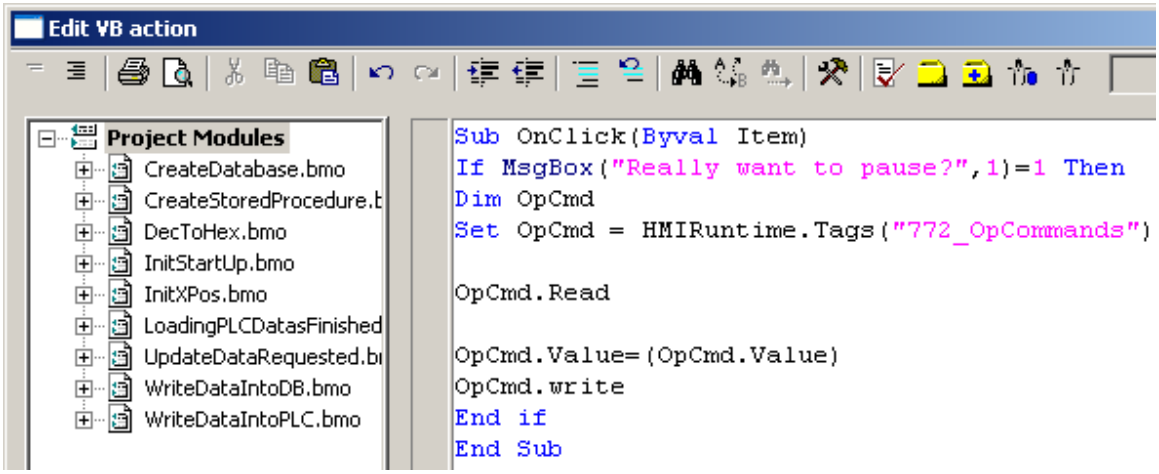


Fig. 3.19

Exemplul 2.

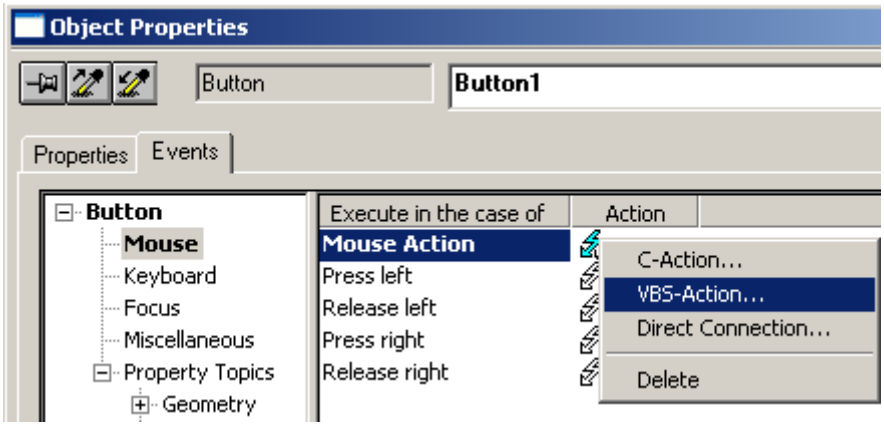


Fig. 3.20

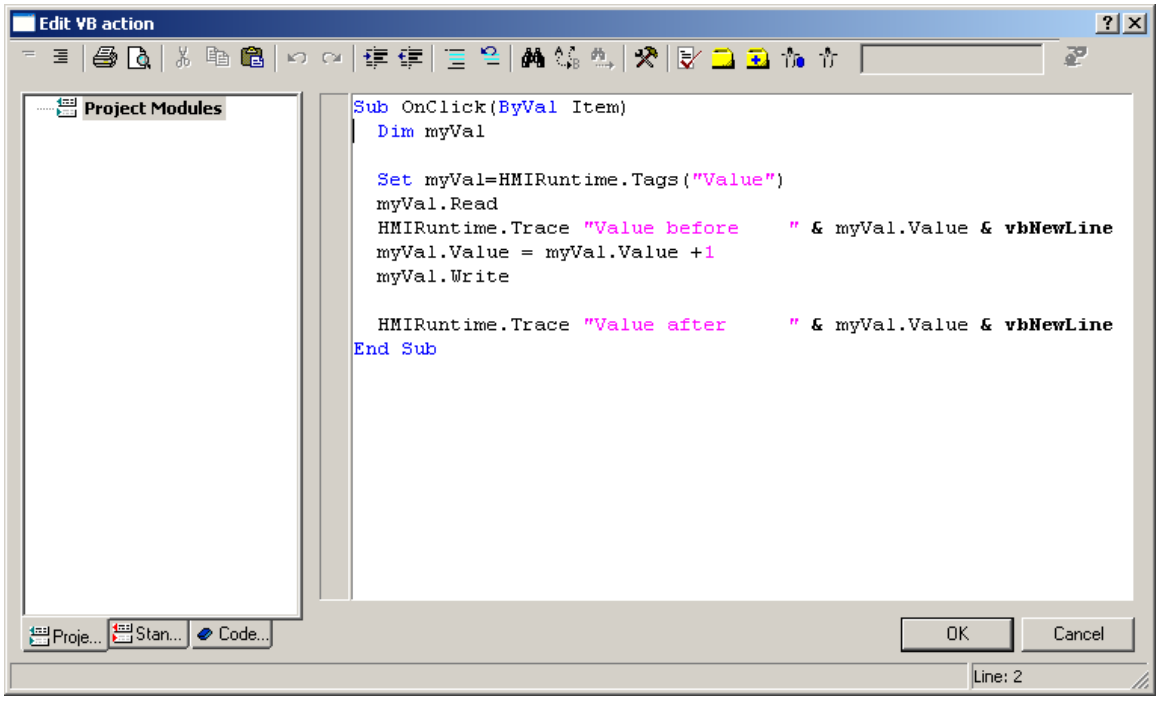


Fig. 3.21

Exemplul 3.

Comanda motor *on/off*:

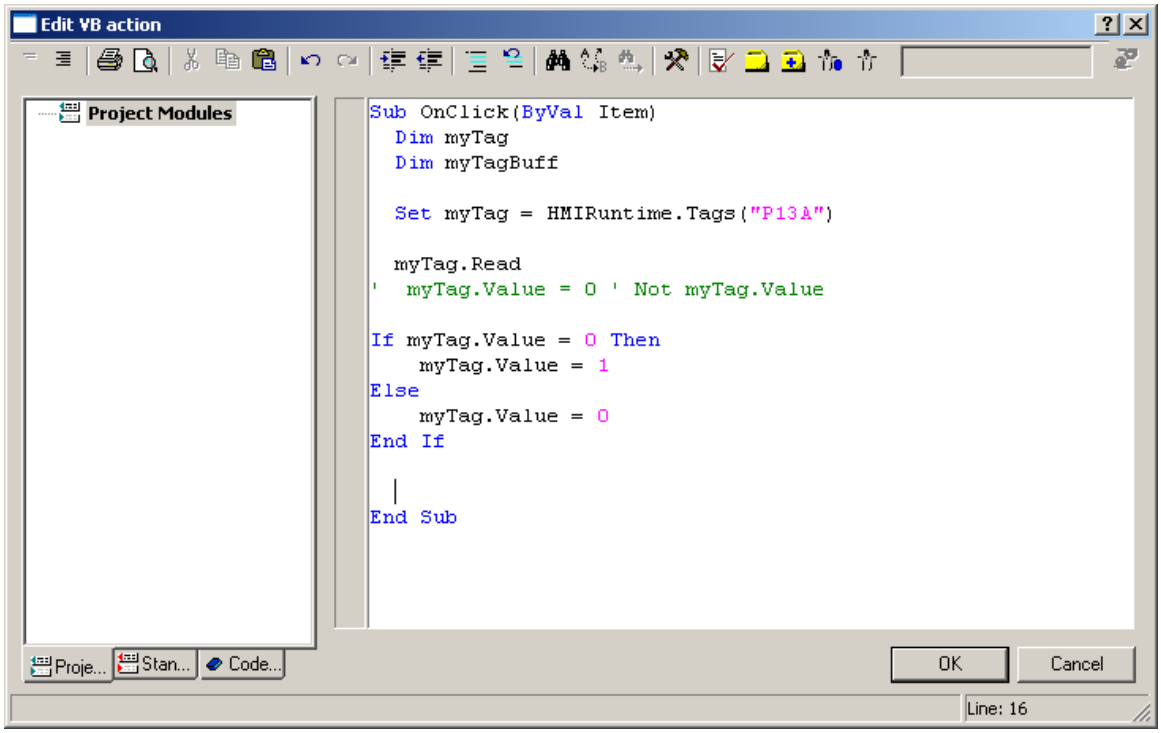


Fig. 3.22

Exemple *C-Action*

Exemplul 1.

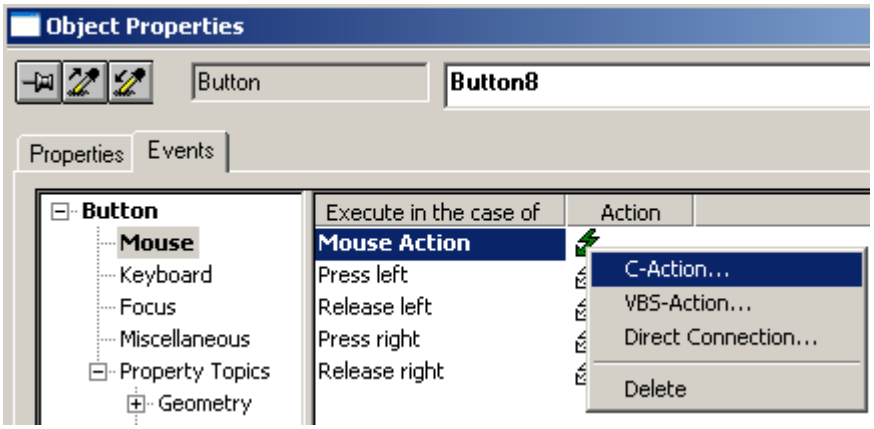


Fig. 3.23

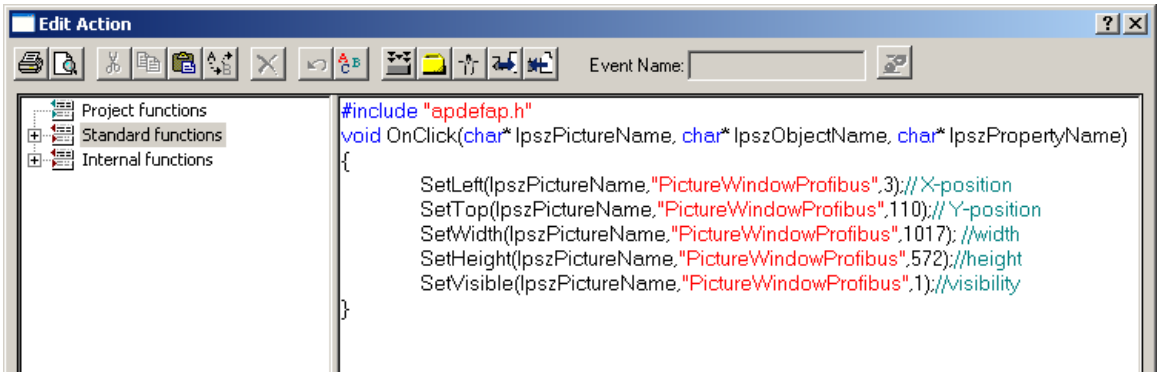


Fig. 3.24

Exemplul 2.

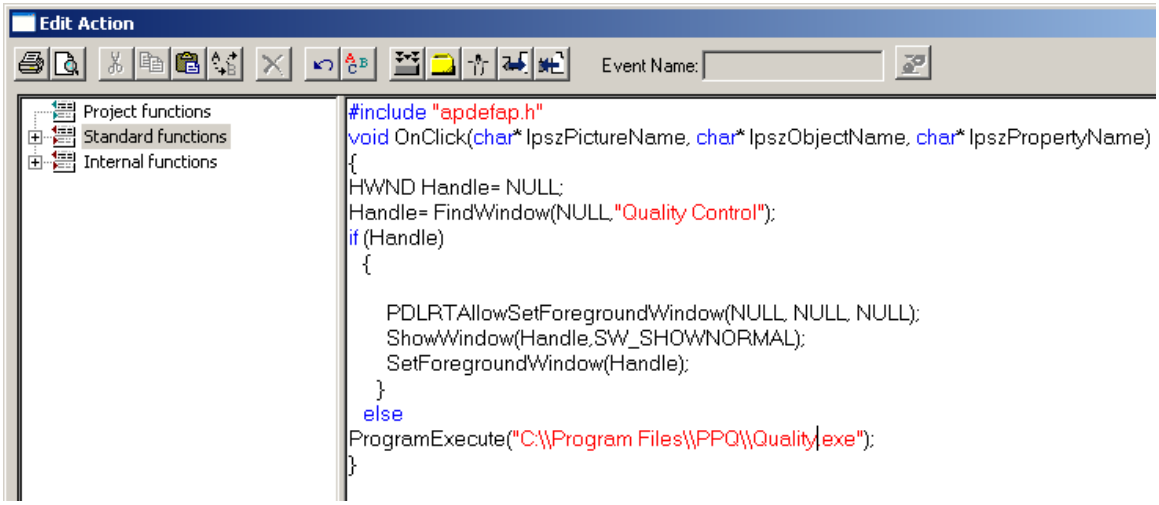


Fig. 3.25

VBS sau C se utilizeaza in cazul in care conexiunile directe cu tagurile nu sunt suficiente, sau mai multe proprietati trebuie modificate in acelasi timp.

Activarea editorului VBA

Cu VBA se largeste functionalitatea Graphics Designer si se automatizeaza configurarea. Graphics Designer are un VBA Editor.

1. *WinCCExplorer/Graphics Designer*

2. *Tools / Macros- ->VisualBasic Editor*

VBA furnizeaza posibilitatea de a automatiza metodele de proiectare/configurare in WinCC Graphics Designer, (crearea de meniuri si toolbars user-defined; crearea si editarea de obiecte standard, Smart si Windows; configurarea de actiuni pentru ecrane/imagini si obiecte etc.

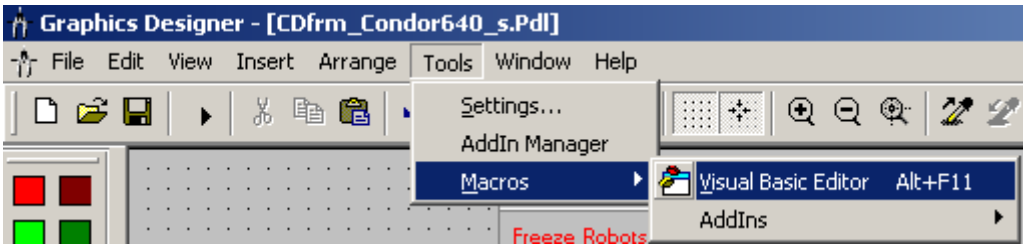


Fig. 3.26

Afisarea informatiilor despre sistem

Canalul System-info-Channel furnizeaza posibilitatea de a defini diferite informatii despre sistem utilizand taguri interne si conectand informatia cu obiecte de intrare si iesire.

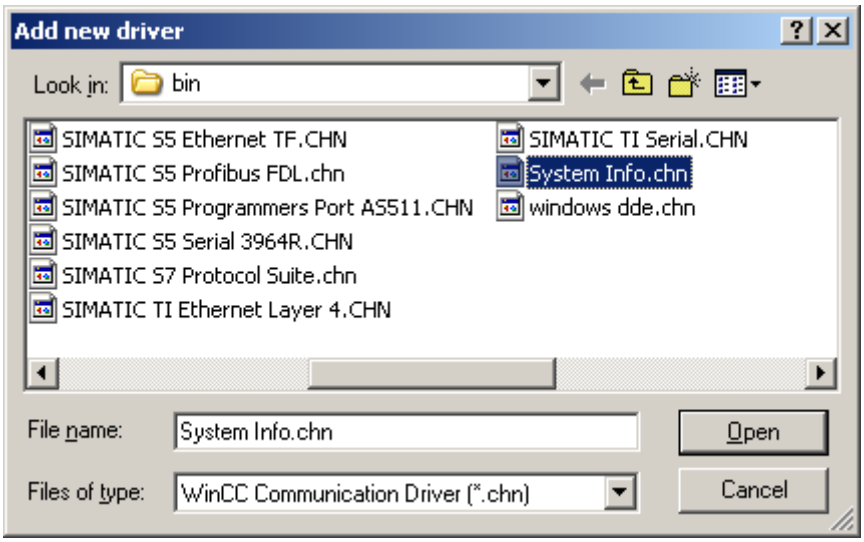


Fig. 3.27

WinCC Explorer / Tag Management / Simatic TI Serial.CHN -> New Tag / Select

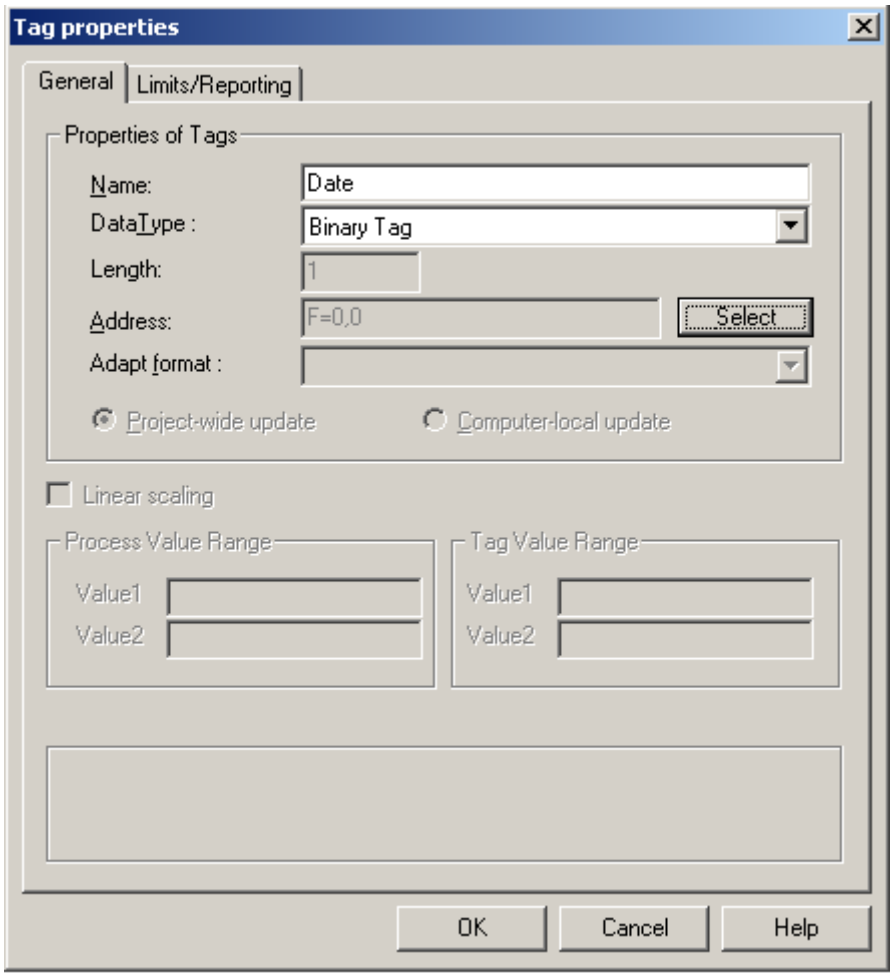

Tag properties
General
Limits/Reporting
Properties of Tags
Name:
Date
DataType :
Binary Tag
Length:
1
Address:
F=0,0
Select
Adapt format :
Project-wide update
Computer-local update
Linear scaling
Process Value Range
Value1
Value2
Tag Value Range
Value1
Value2
OK
Cancel
Help

Fig. 3.28

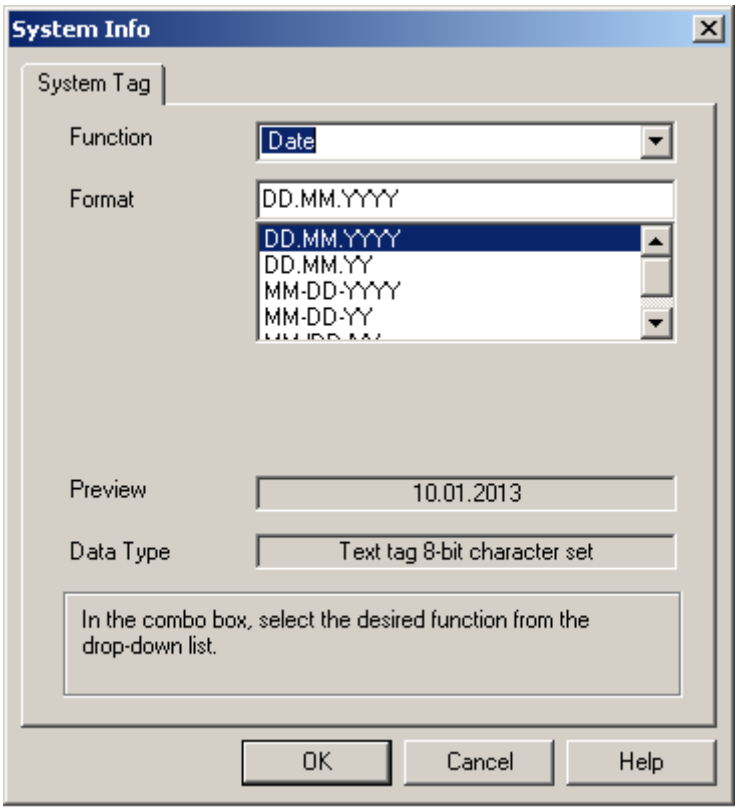


Fig. 3.29

Exemple

Exemplul 1.

Afisarea temperaturii cu ajutorul obiectului *bar* din *Smart Objects*.

Selecteaza un obiect *bar* din grupul *Smart Objects*. Se configureaza cu *Configuration Dialog*. Se defineste domeniul [0…100].

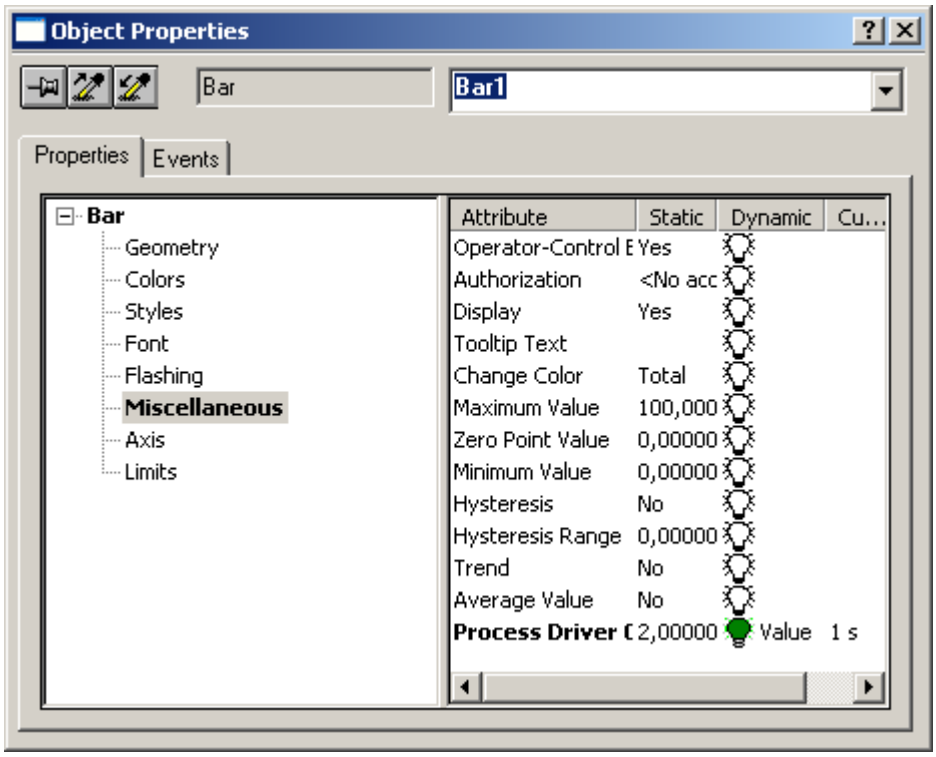


Fig. 3.30

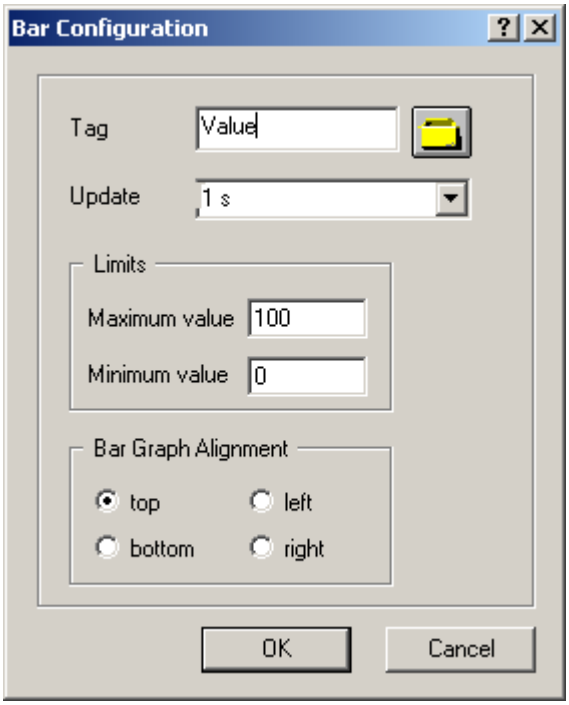


Fig. 3.31

Modificarea de culoare la depasirea limitelor poate fi configurata prin proprietatea "*bar color*" si Dynamic Dialog.

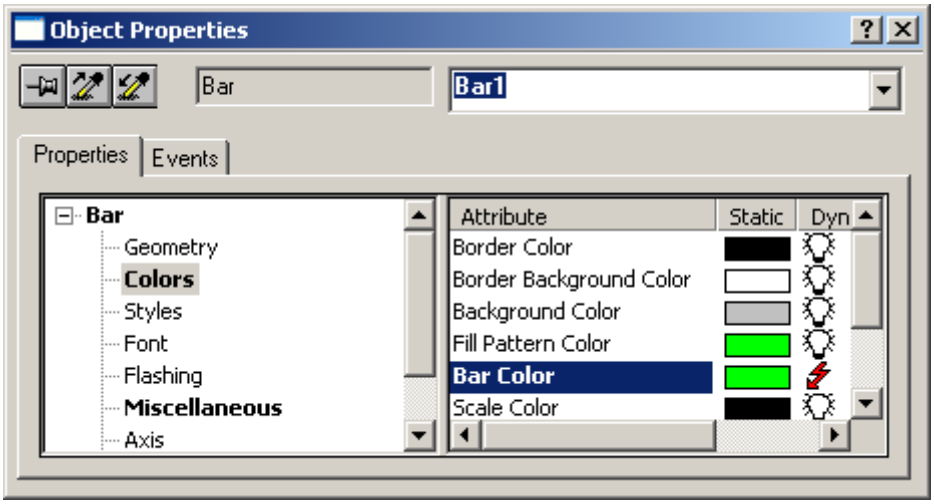


Fig. 3.32

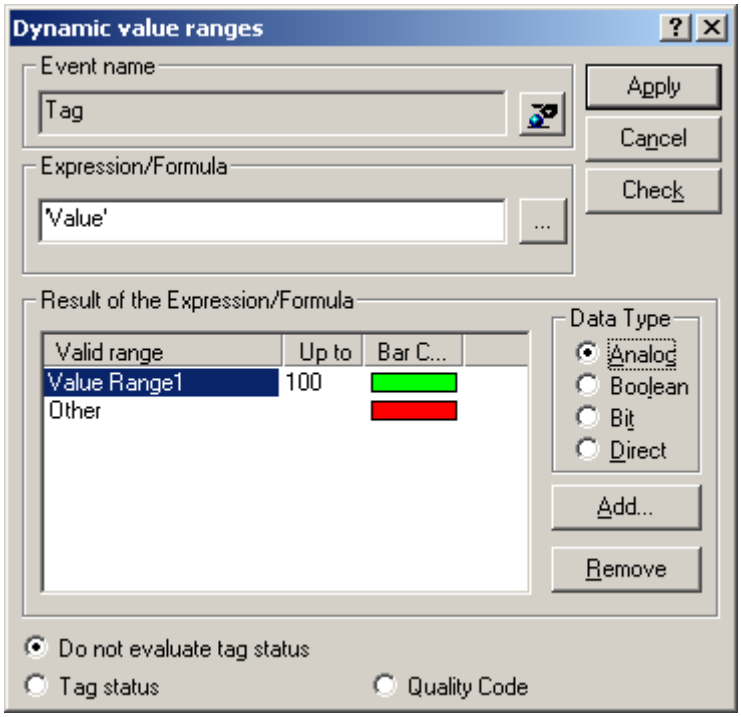


Fig. 3.33

Exemplul 2:

Crearea unui ecran de start

Se creaza 2 ecrane '*Start.pdl*' si '*Motor.pdl*' .

Din acest ecran este posibila selectia ecranului de start.

1. Se configureaza 2 ecrane:

   a. Ecranul de Start : *Start.PDL*. Din acest ecran este posibila selectia ecranului *Motor.PDL*.

   b. Ecranul Motor: *Motor.PDL*. Din acest ecran este posibila selectia ecranului de start.

Configureaza un buton pentru functia de selectie ecran, si cate un *Picture Window*

2. Configureaza ambele functii de selectie ecran (comuta la *Motor.PDL* si inapoi).
a. Exemplu cu *C-Action*

Buton/Properties/Events/Mouse Action/C-Action

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName,
char* lpszPropertyName)

{

SetLeft(„Start.PDL","PictureWindow107",0);// X-position
SetTop(„Start.PDL","PictureWindow107",109);// Y-position
SetWidth(„Start.PDL","PictureWindow107",1024); //width
SetHeight(„Start.PDL","PictureWindow107",570);//height
SetVisible(„Start.PDL","PictureWindow107",1);//visibility

}
```

b. Exemplu cu *Direct Connection*

Buton/Properties/Events/Mouse Action/Direct Connection

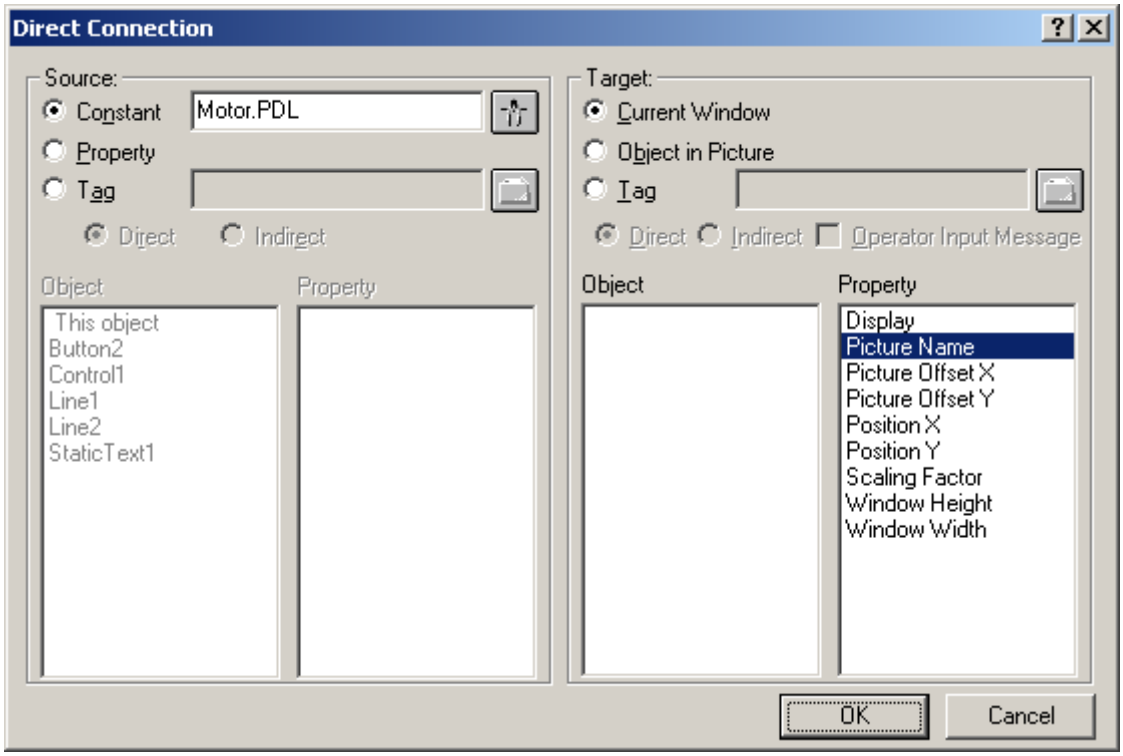


Fig. 3.34

3. In ecranul *Start.PDL* configureaza un buton de *Exit Project & Runtime* configurat cu *Dinamic Wizard*.

4. Salveaza ecranul.

5. Selecteaza *Start.PDL* ca ecran de start din modul *Runtime*.

6. Salveaza.

Exemplul 3:

Configurarea posibilitatilor de schimbare a culorii

Pentru schimbarea culorii obiectelor exista mai multe posibilitati:

1. *Dynamic Dialog*

2. *C action*

3. Conectarea *tag*-urilor la *property/color* corespunzatoare.

Aplicatii: Modificarea culorii unui obiect este folosita pentru stare semnal, cum ar fi:

- Stare echipament

- Depasirea valorilor limita

- Starea conexiunilor cu procesul

Exemplul 4:

Afisarea starii unui motor *on/off*.

Un obiect *Static text* este utilizat pentru afisarea starii *on/off* (*Graphics Designer / Object Palette / Static text*).

Starea "*off*" este definita in culoarea gri si textul “OFF”, si starea "*on*" este definita in culoarea verde si textul “ON”.

Static text/ Properties/ Styles/ Fill Pattern/ Dynamic Dialog

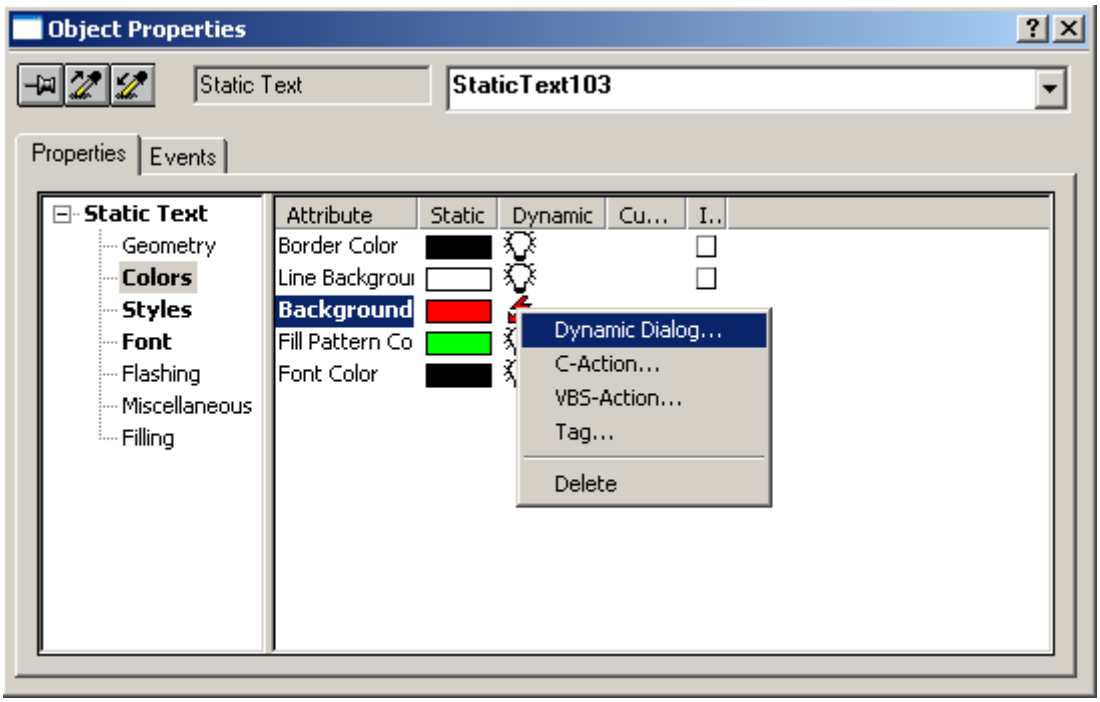


Fig. 3.35

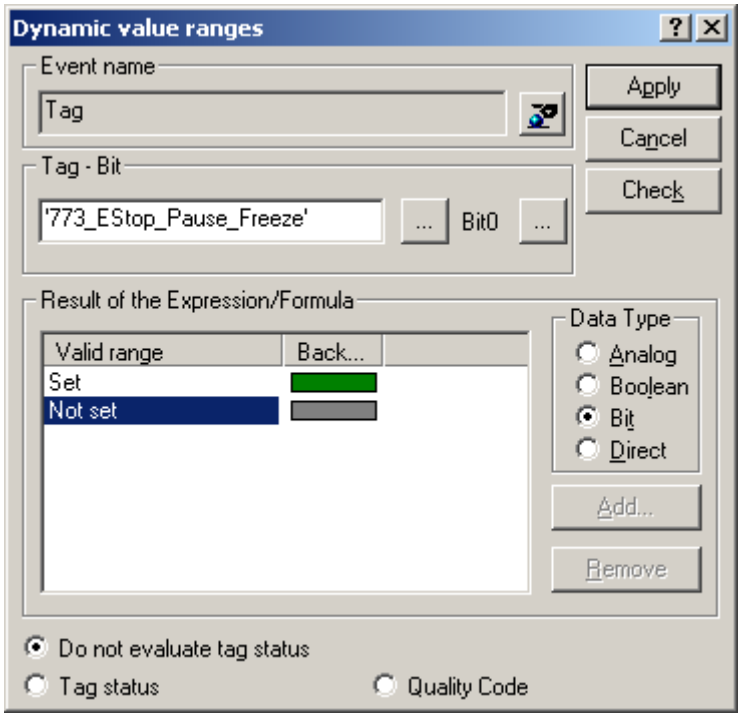


Fig. 3.36

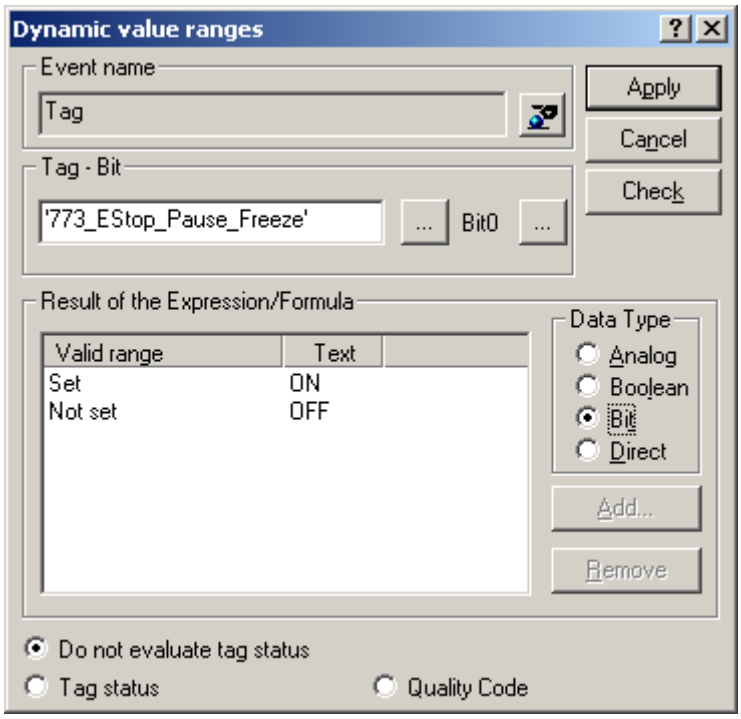


Fig. 3.37

Exemplul 5:

Semnalarea depasirii limitelor.

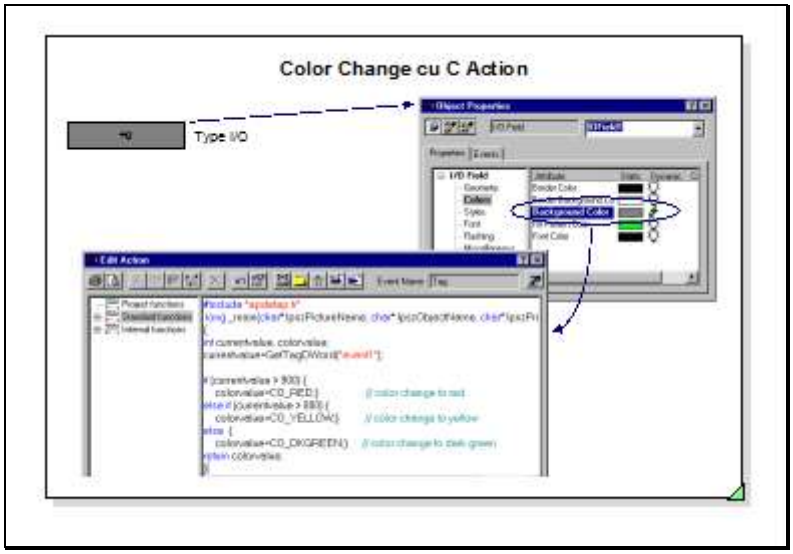


Fig. 3.38

Un tag de proces este folosit pentru a semnala depasirea limitelor. Depasirea limitelor este vizibila prin modificarea culorii. Tagul accepta valori intre 0 si 100.

Se configureaza:

Value > 80: color animation from dark green to yellow

Value > 90: color animation from yellow to red

Se utilizeaza o componenta I/O Field. *Graphics Designer/Object Palette/I/O Field / Properties/Colors/Background Colors/ C – Action.*

```
int currentvalue, colorvalue;
currentvalue = GetTagDWord("event1");
if (currentvalue > 900) {
  colorvalue = CO_RED; }                                    //
color change to red
else if (currentvalue > 800) {
  colorvalue = CO_YELLOW; }                          //      color
change to yellow
else {
  colorvalue = CO_DKGREEN; }                         //      color
change to dark green
return colorvalue;
```

Operatii On/Off : Exista cateva posibilitati pentru comutarea *on* / *off*.

1. Direct Connection

2. Dynamic Wizard

3. C-Action

Cea mai buna solutie pentru sistem este Direct Connection, deci se recomanda utilizarea acesteia daca este posibil.

Exemplul 6:

Comutarea on si off cu confirmare. Se configureaza 3 butoane:

Butonul *On/Off*: Object Properties/Events / Mouse Action/VBS-Action

```
Sub OnClick(ByVal Item)
  Dim myTag
  Dim myTagBuff

  Set myTag = HMIRuntime.Tags("P13A")

  myTag.Read
'  myTag.Value = 0 ' Not myTag.Value

If myTag.Value = 0 Then
     myTag.Value = 1
Else
     myTag.Value = 0
End If

  Set myTagBuff = HMIRuntime.Tags("P13A_Buff")
  myTagBuff.Read
  myTagBuff.Value = myTag.Value
  myTagBuff.Write

End Sub
```

Butonul *Confirmare*: Object Properties/Events / Mouse Action/Direct-Connection

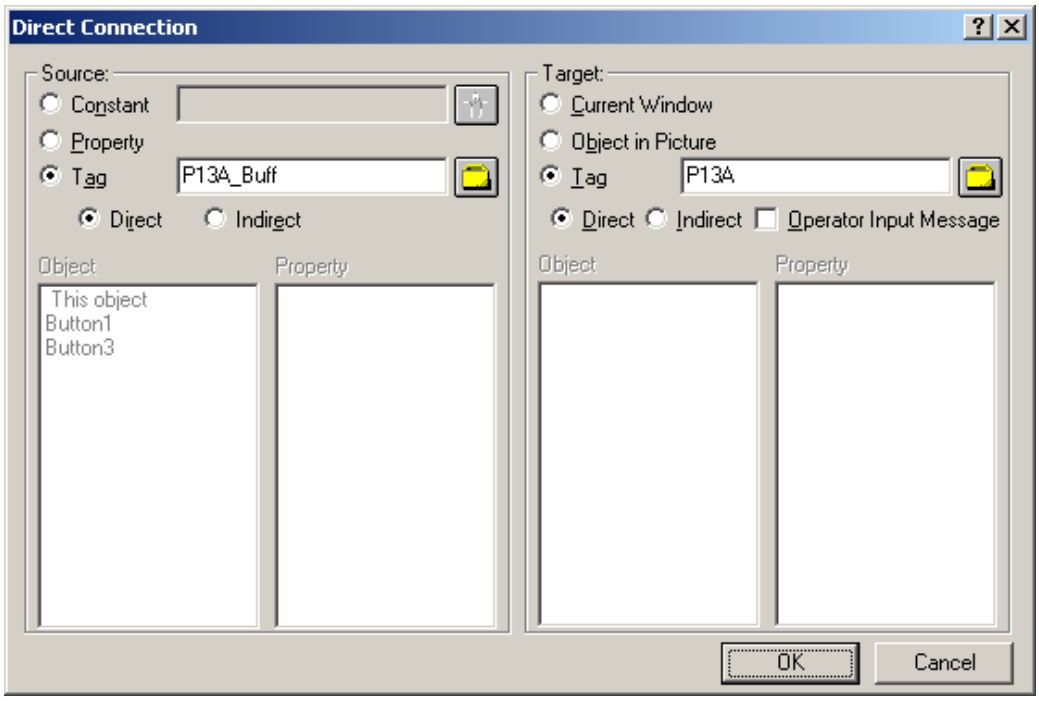


Fig. 3.39

Butonul *OK*: Object Properties/Events / Mouse Action/Direct-Connection

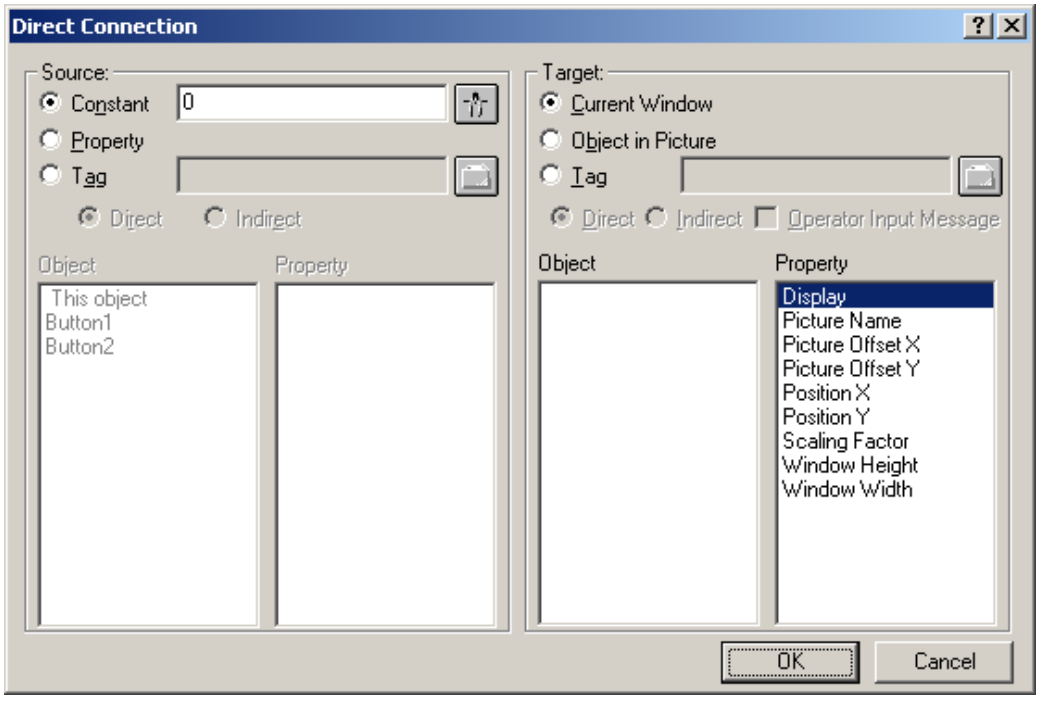


Fig. 3.40

Butonul *OK* se utilizeaza pentru inchiderea ecranului.

Exemplul 7:

Configureaza 2 butoane '*Referinta'* pentru setarea referintei cu valorile 100 respectiv 150, un buton de '*Enable'* , un obiect Static text si un *I/O field* pentru afisarea valorii.

Obiectul Static text afiseaza initial textul: 'Enable missing!' (*Properties/Font/Text*)

La apasarea butonului '*Enable*' mesajul afisat in Static text dispare *(SetVisible(lpszPictureName,"StaticText1",0))*, se permite modificarea referintei cu unul din cele doua butoane

(SetPropBOOL(lpszPictureName,"Button1","Operation",TRUE)).

```
#include "apdefap.h"
void  OnClick(char*  lpszPictureName,  char*  lpszObjectName,
char* lpszPropertyName)

{

// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
#define ApcEnable  "Enable"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
// WINCC:PICNAME_SECTION_END

BOOL ok;
ok=GetTagBit(ApcEnable);    //Return-Type: BOOL

ok=!ok;
SetTagBit(ApcEnable,(WORD)ok);   //Return-Type: BOOL
SetVisible(lpszPictureName,"StaticText1",0);
if (ok==FALSE){
```

```
SetPropChar(lpszPictureName,"Button1","ToolTipText","Not
Enabled"); //Return-Type: BOOL
SetPropChar(lpszPictureName,"Button2","ToolTipText","Not
Enabled"); //Return-Type: BOOL
SetPropBOOL(lpszPictureName,"Button1","Operation",FALSE);
     //Return-Type: BOOL
SetPropBOOL(lpszPictureName,"Button2","Operation",FALSE);
     //Return-Type: BOOL
}
else
{
SetPropChar(lpszPictureName,"Button1","ToolTipText","Enabled")
;    //Return-Type: BOOL
SetPropChar(lpszPictureName,"Button2","ToolTipText","Enabled")
;    //Return-Type: BOOL
SetPropBOOL(lpszPictureName,"Button1","Operation",TRUE);  //Return-Type:
BOOL
SetPropBOOL(lpszPictureName,"Button2","Operation",TRUE);
     //Return-Type: BOOL

}

}
```

La apasarea butoanelor de setare referinta se inscriu in Tag-ul *Value* valorile 100 respectiv 150.

```
#include "apdefap.h"
void  OnClick(char*  lpszPictureName,  char*  lpszObjectName,
char* lpszPropertyName)

{

// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
#define ApcEnable  "Enable"
#define ApcValue  "Value"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
// WINCC:PICNAME_SECTION_END

BOOL ok;
ok=GetTagBit(ApcEnable);    //Return-Type: BOOL
```

```
if (ok==FALSE)
  SetVisible(lpszPictureName,"StaticText1",1);
else
  SetTagWord(ApcValue,100);//Return-Type: BOOL

}
```

Valoarea referintei este afisata in *I/O Field*: *Properties/Output/Input/Output Value/Tag.*

Exercitiul 1:

Sa se configureze un ecran care sa permita setarea referintei cu valorile 100 si 150. In acest scop se vor utiliza 2 *butoane* care se vor configura cu *Direct Connection*. Valoarea referintei se va afisa intr-un *I/O Field*.

Exercitiul 2:

Folositi un obiect *Slider* (*Graphics Designer/Object Palette/Windows Objects/Slider Object*) pentru a introduce o referinta. Setati domeniul [0..100], *Process Connection / Tag.*

Exercitiul 3:

Se poate folosi un obiect *check box* pentru a face multiple selectii. Sa se afiseze cate un obiect din tangram pentru fiecare camp selectat cu ajutorul unui *check box*.

Obiectul *Check box* se va configura cu tag connection: *Properties/Output Input/Selected Boxes/Tangram* (Signed 32 32-bit value).

Obiectele din tangram se configureaza cu *Polygon/ Properties/Misc/Display/Dynamic Dialog/Tangram: (pentru selectii se pot utiliza biti)*

Bit 0

Bit 1

Bit 2

Bit 3

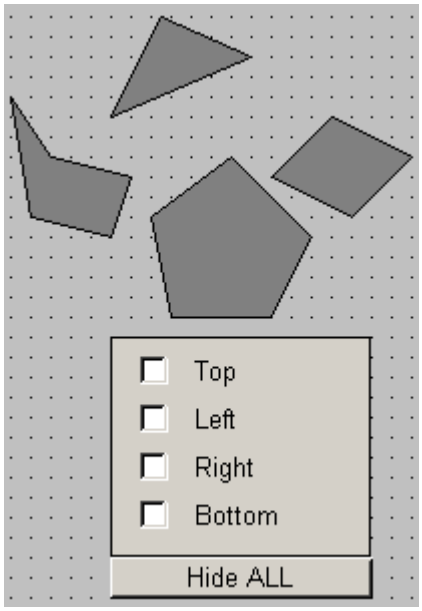


Fig. 3.41

Butonul Hide ALL se configureaza cu Direct Connection:

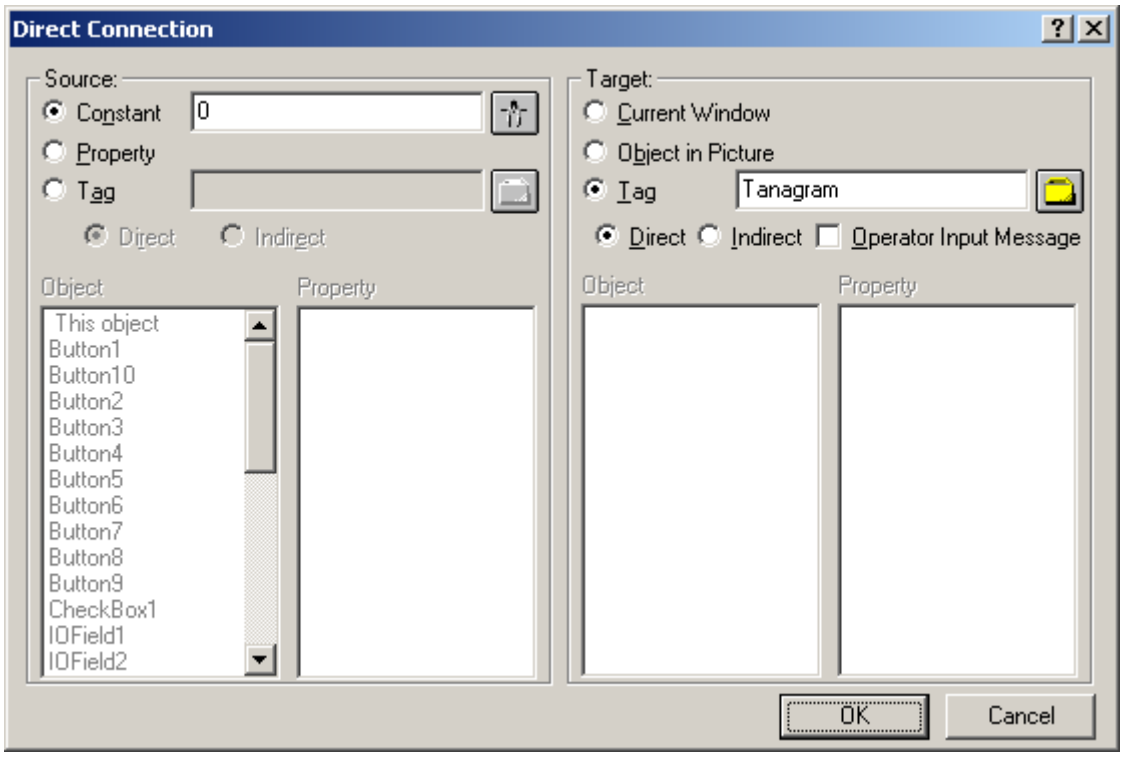


Fig. 3.42

Adresarea indirecta

Cu adresarea indirecta a atributelor, numarul de obiecte si dinamici asociate se poate reduce si deasemenea continutul ecranului e mai usor de citit.

Exemplu:

Nivelul a 2 rezervoare trebuie reprezentat printr-un camp de iesire.

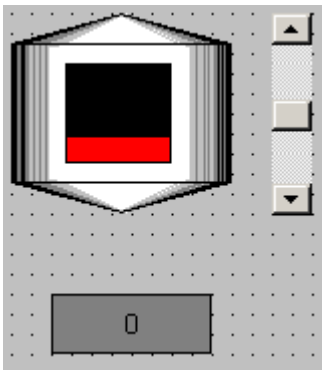


Fig. 3.43

1. Se creaza un tag intern "*textvariable*" de tipul "*Text tag 8-bit character set*"

2. Se creaza obiecte de configurare ca in exemplu.

3. Asignarea tagului intern "*textvariable*" se face prin *Direct Connection - mouse click*. Parametrul sursa este transmis parametrului referinta.

4. Astfel valoarea tagului "*I_Value_1*" este transmis la I/O Field utilizand atributul "*Output Value*".

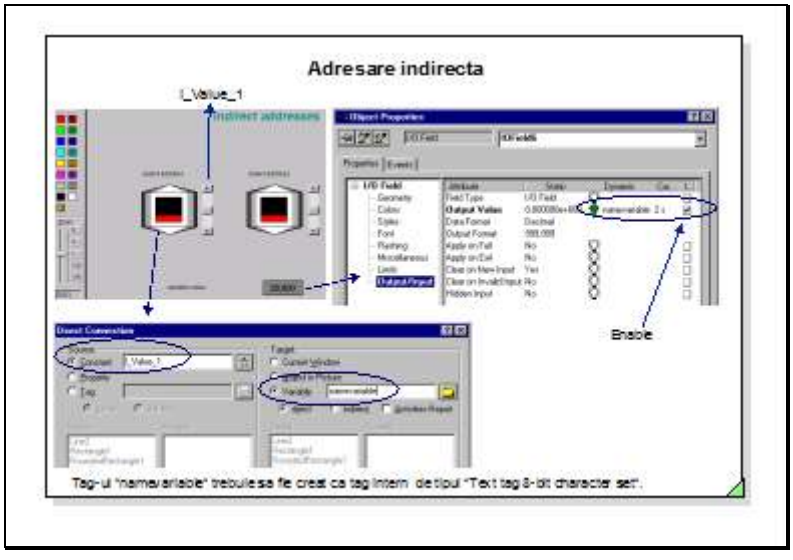


Fig. 3.44

Tag-ul “namevariable” trebuie sa fie creat ca tag intern de tipul “Text tag 8 –bit character set”.

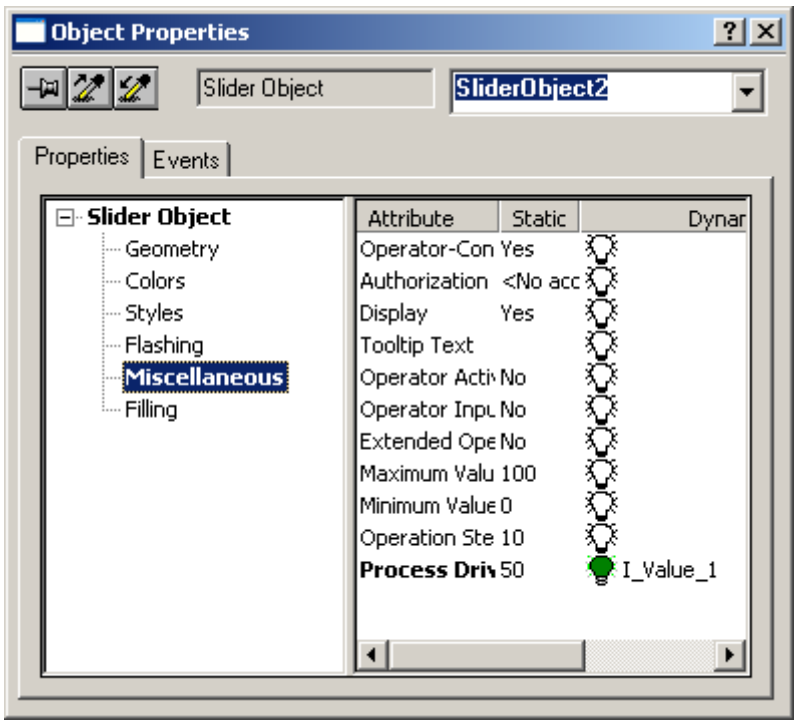


Fig. 3.45

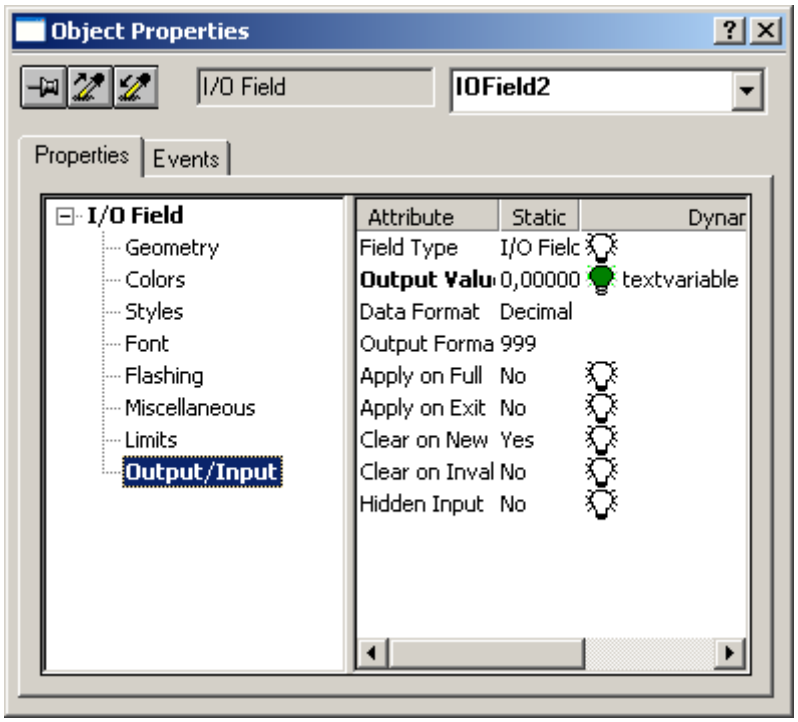


Fig. 3.46

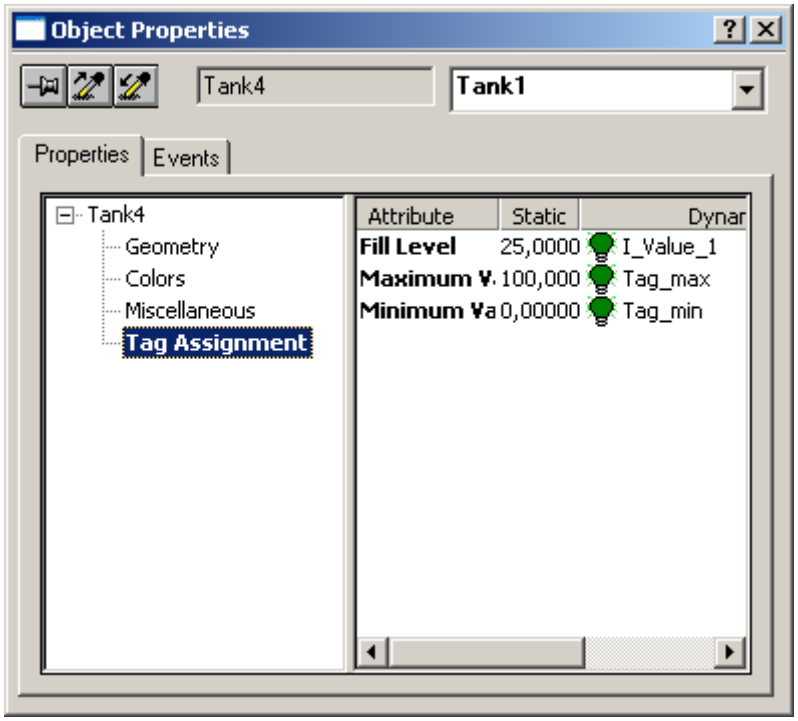


Fig. 3.47

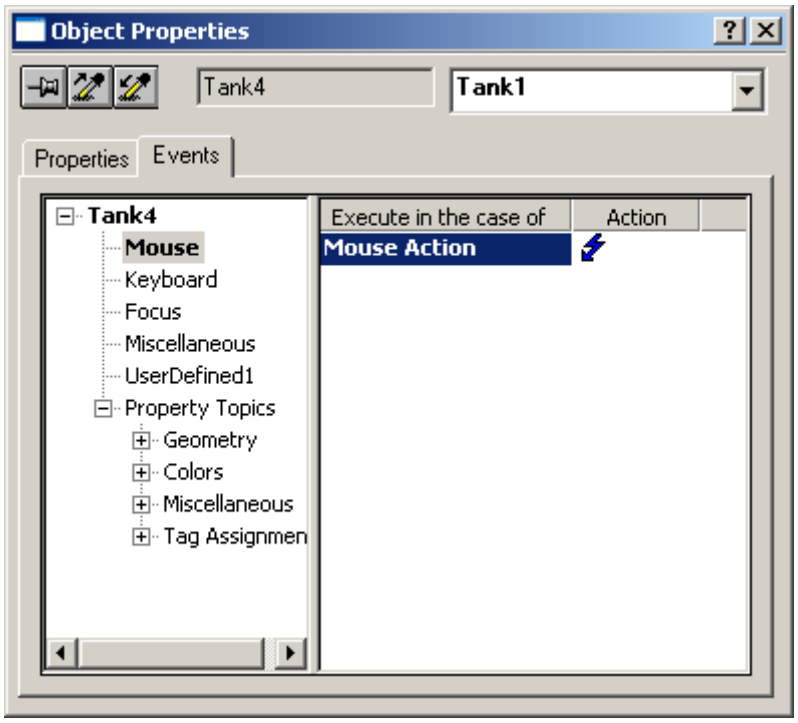


Fig. 3.48

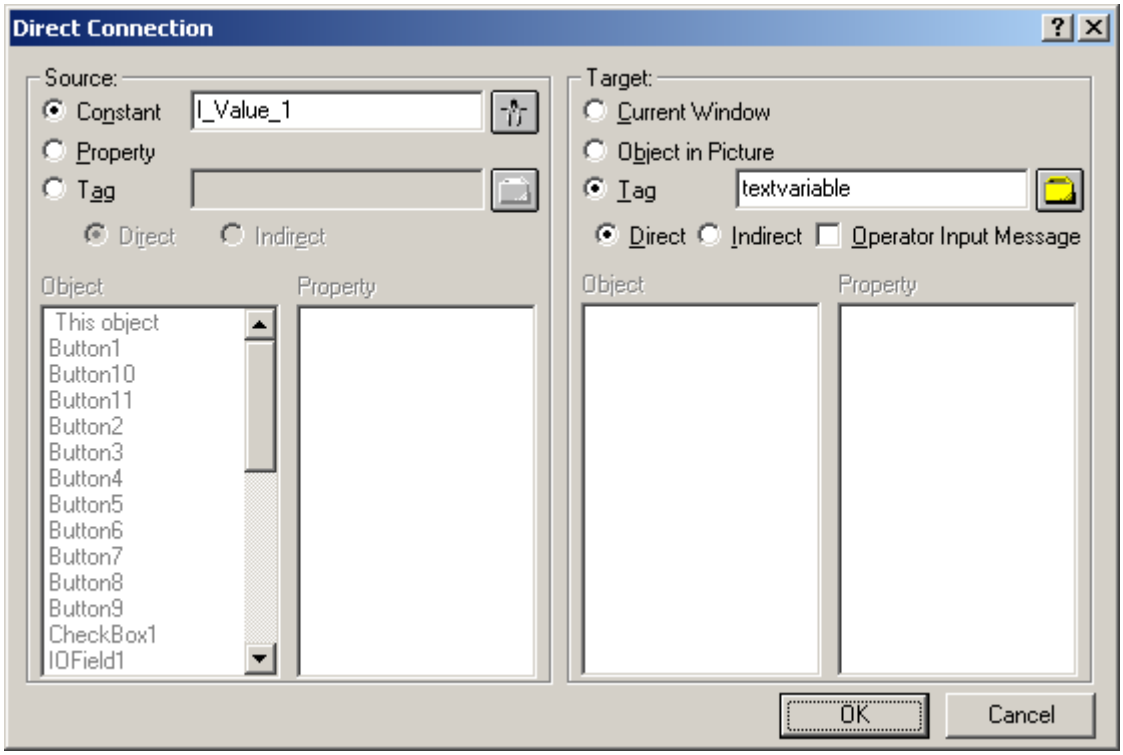


Fig. 3.49

Block Technology

Tehnica *Block Technology* permite crearea de functii (cum ar fi blocul motor). Blocul creat se pote utiliza de multe ori, si de cate ori e chemat blocul se apeleaza cu propriile date.

Pentru a reduce costurile de configurare si management ale obiectelor de proces similare, WinCC ofera posibilitatea de utilizare a tehnologiei *picture block*.

Pe baza similaritatii structurale a datelor inregistrate in PLC, se creaza *Structure Tag* (de exemplu pentru blocul motor, etc). Acestea reprezinta interfata utilizata pentru a citi/scrie date in PLC de la un anumit obiect specific.

Exemplu:

Definirea unui bloc ‘ComandaMotor’ cu posibilitatea de comanda On/Off. Blocul este apelat de 3 ori in ecranul Motor.

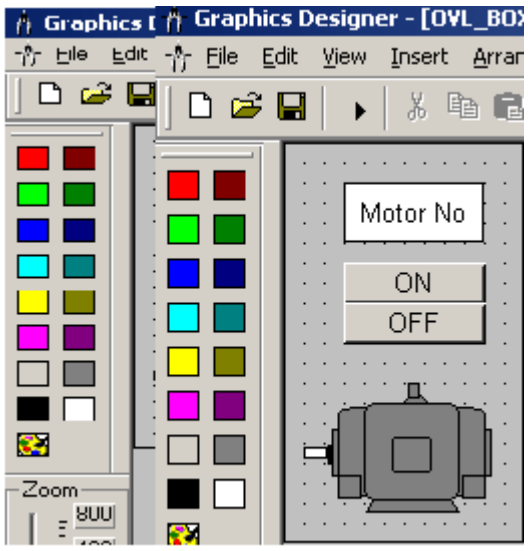


Fig. 3.50

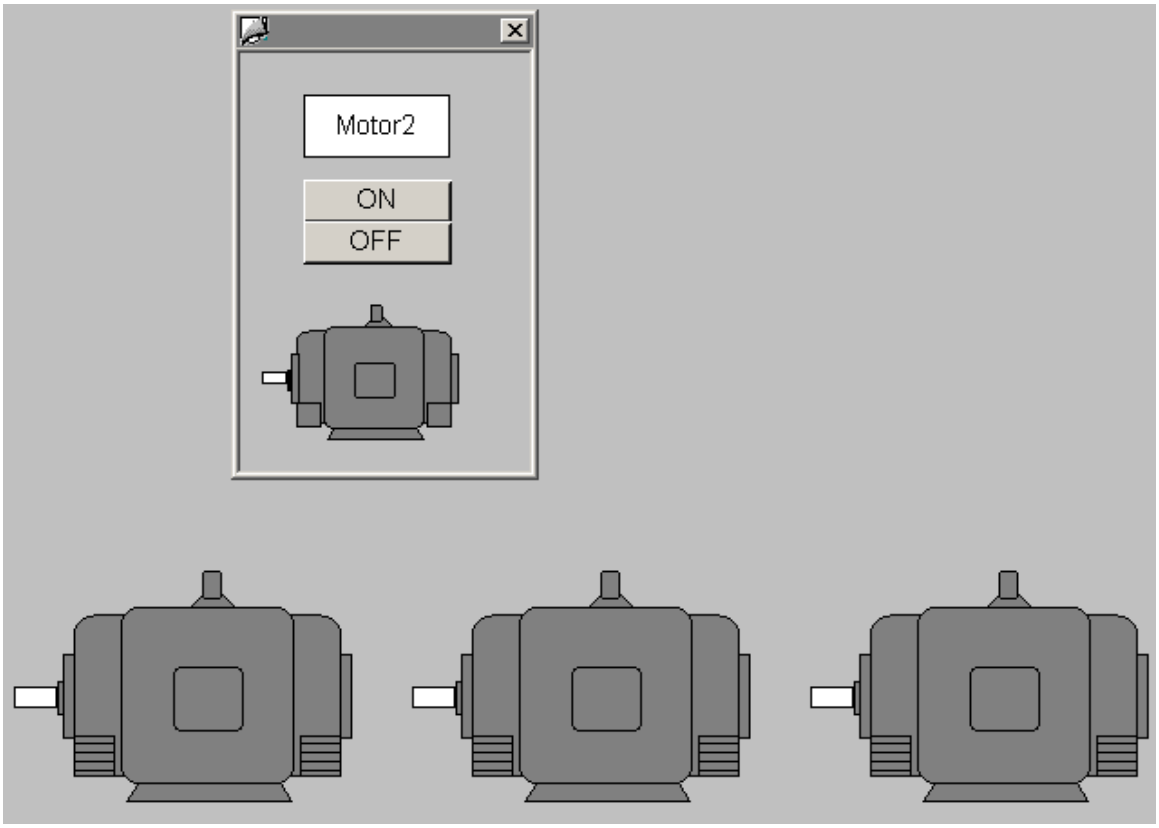


Fig. 3.51

Se creaza un ecran *'Motor.pdl'*. Se insereaza 3 obiecte grafice din *View/Library/Plant elements/Motors*. Fiecare obiect *Motor* isi schimba culoarea *(Properties/Color/Background/Dynamic Dialog)*

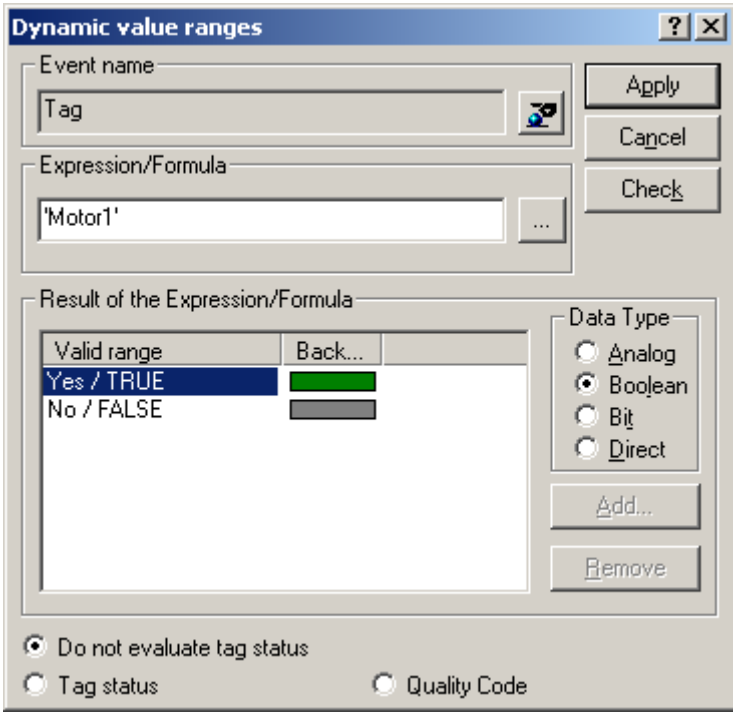


Fig. 3.52

Pentru fiecare obiect de tip Motor se defineste un eveniment la mouse click pe obiect care deschide ecranul 'ComandaMotor'. (*Events/Mouse/Mouse Action/C-Action*).

In acest scop se insereaza in ecranul Motor un obiect Picture Window (*PictureWindow1*)

```
#include "apdefap.h"
void OnClick(char* lpszPictureName, char* lpszObjectName, char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
#define ApcI_VAR_MOT_NO  "I_VAR_MOT_NO"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
#define ApcComandaMotor_Pdl  "ComandaMotor.Pdl"
// WINCC:PICNAME_SECTION_END

SetTagChar(ApcI_VAR_MOT_NO, "Motor2");
```

```
SetPictureName(lpszPictureName,"PictureWindow1",ApcComandaMoto
r_Pdl);     //Return-Type: BOOL
SetVisible(lpszPictureName,"PictureWindow1",1);   //Return-
Type: BOOL

}
```

In ecranul ComandaMotor.pdl se insereaza un obiect Static Text, 2 obiecte Buton si un obiect motor (View/Library).

Obiectul *Static Text:*

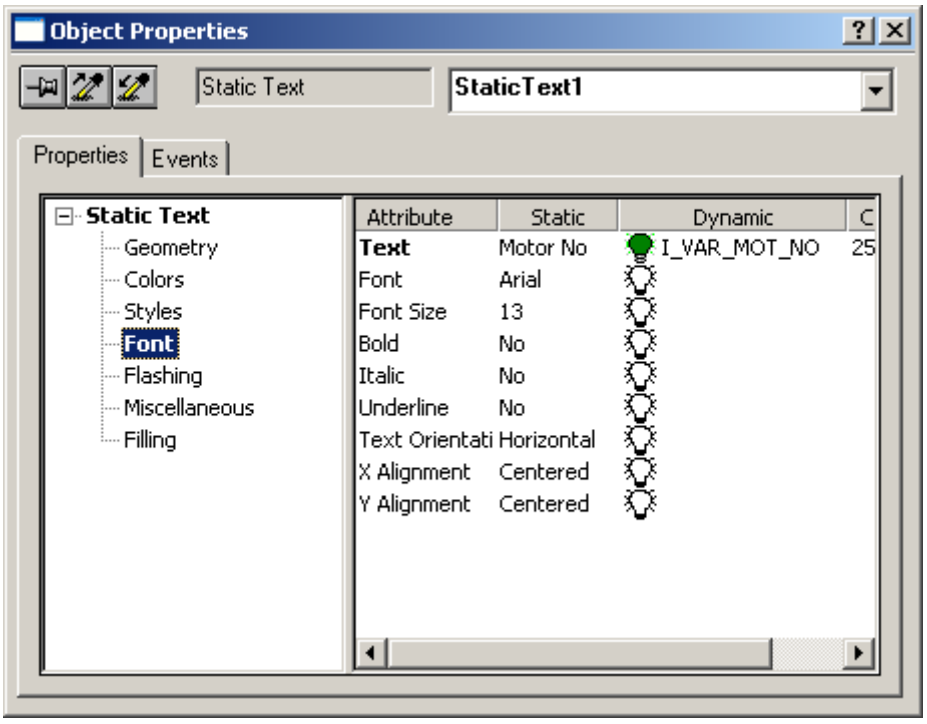


Fig. 3.53

Pentru obiectele *Buton*:

Butonul ON se configureaza cu *Event/Mouse Action/C-Action*:

```
#include "apdefap.h"
void  OnClick(char*  lpszPictureName,  char*  lpszObjectName,
char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
#define ApcI_VAR_MOT_NO  "I_VAR_MOT_NO"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
// WINCC:PICNAME_SECTION_END
SetTagBit(GetTagChar(ApcI_VAR_MOT_NO),1);
}
```

Butonul OFF se configureaza cu *Event/Mouse Action/C-Action*:

```
#include "apdefap.h"
void  OnClick(char*  lpszPictureName,  char*  lpszObjectName,
char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
#define ApcI_VAR_MOT_NO  "I_VAR_MOT_NO"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
// WINCC:PICNAME_SECTION_END
SetTagBit(GetTagChar(ApcI_VAR_MOT_NO),0);
}
```

Pentru obiectul Motor, se modifica culoarea in functie de starea ON/OFF cu *Properties/Color/Background/C-Action*

```
#include "apdefap.h"
 long _main(char* lpszPictureName, char* lpszObjectName, char*
lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction DMTagName
// next TagID : 1
```

```
#define ApcI_VAR_MOT_NO  "I_VAR_MOT_NO"
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction PictureName
// next PicID : 1
// WINCC:PICNAME_SECTION_END
BOOL x;
x=GetTagBit(GetTagChar(ApcI_VAR_MOT_NO));
if (x)
  return CO_DKGREEN;
else
  return CO_DKGRAY;
}
```

User Administrator

Asignarea si managementul drepturilor de acces poate fi configurata cu *WinCCExplorer/User Administrator*. Fiecare comanda de operare in proces, arhivare si operare sistem WinCC poate fi protejata impotriva accesului neautorizat. Exemple de operatii care pot fi protejate impotriva accesului neautorizat includ: modificari de referinta, accesul la ecrane, accesarea software-ului de configurare de catre operatorii de proces, etc. Exista cateva nivele de acces care permit configurarea unei protectii ierarhizate, autorizatii exclusive de operare pentru operatorii individuali.

Exemplu

O ajustare de referinta si chemarea unui ecran de proces pentru a introduce un ordin de productie este protejata impotriva accesului neautorizat. Selectia ecranelor si ajustarea referintei este permisa de catre un anume utilizator "Utilizator1", utilizand " Utilizator1" login si parola " Utilizator1_MS". Operatorul "Utilizator2" utilizeaza" Utilizator2" login si parola " Utilizator2" pentru a vizualiza ordinele dar nu ii este permis sa ajusteze referinta.

Configurare: Se definesc *hot keys* pentru *logon* si *logoff*. (*Proiect/Project properties/ Hotkeys*)

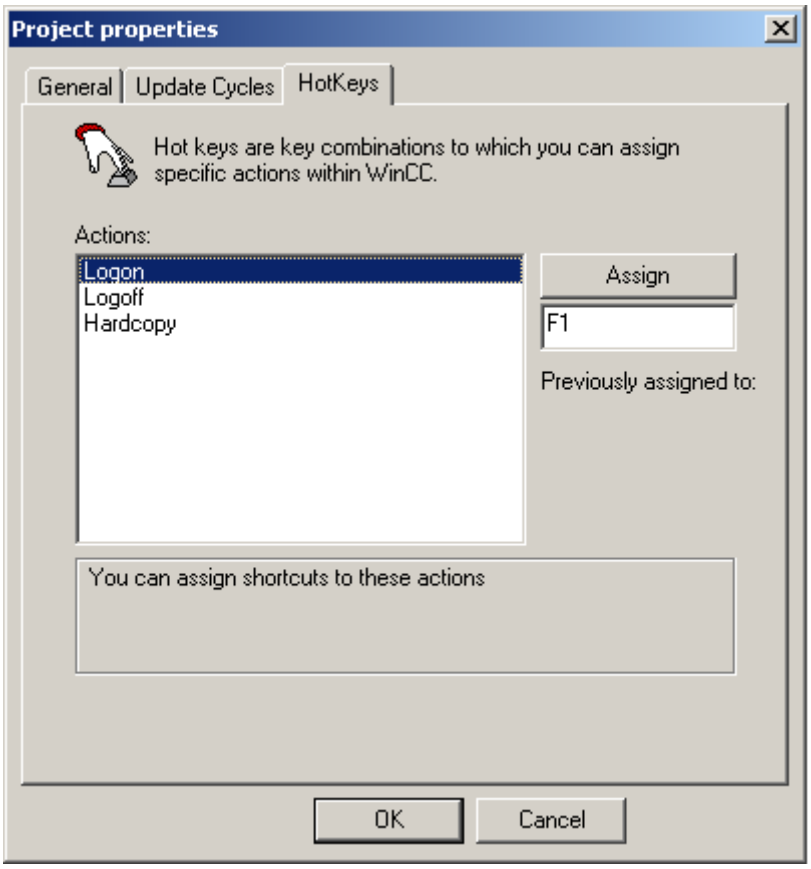


Fig. 3.54

Logon – cand se utilizeaza F1. Se introduce numele si parola astfel incat utilizatorul poate efectua operatii protejate de parola.

Logoff - cand se utilizeaza F2.

User Administrator Editor este utilizat pentru a introduce numele si parola de login. Porneste *User Administrator*.

Se pot configura autorizatii utilizator si autorizatii de grup.

Crearea unui Group

Din meniu*"User/Add Group"*.

Crearea unui User

Din meniu*"User/Add User"*.

Se vor defini pentru fiecare utilizator nivelele de autorizare (dublu click pe coloana "*Authorization*").

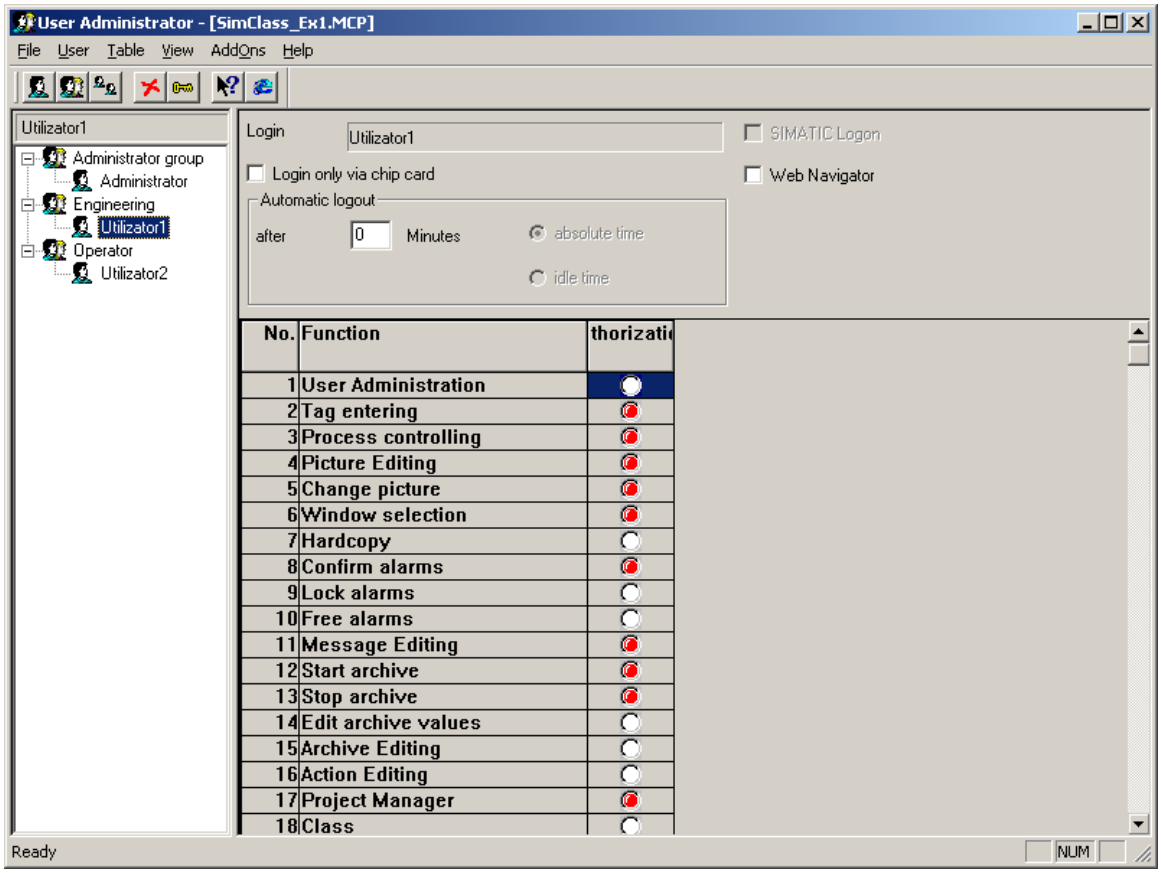


Fig. 3.55

In ecranele individuale se selecteaza obiectele pe care dorim sa le protejam cu parola si se selecteaza nivelul de autorizare - *Authorization* asa cum e descris in *User Administrator*.

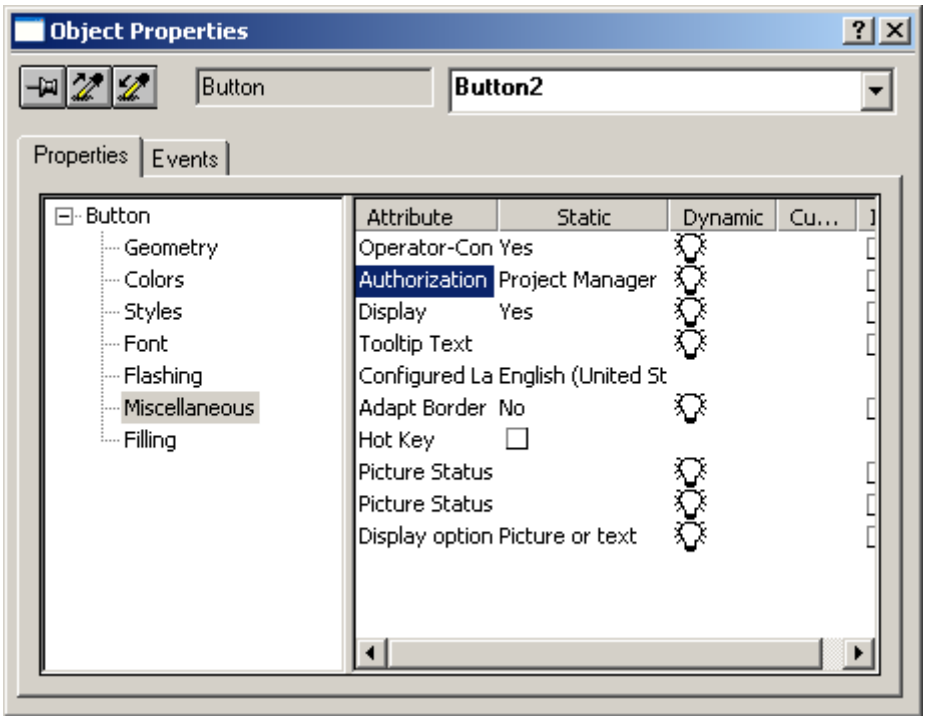


Fig. 3.56

Se reactiveaza runtime dupa configurarea parolei.

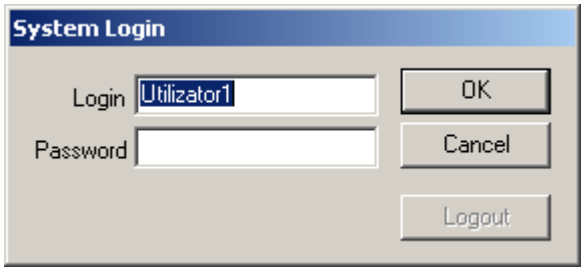


Fig. 3.57

Exercitiu:

Configureaza pasii descrisi anterior pentru definirea de grupuri/ utilizatori si defineste in ecranele Start.pdl si Motor.pdl obiectele si nivelele de autorizare.

Capitolul 4. Afisarea si arhivarea mesajelor

4.1. Configurare

Prin sistemul de mesaje, evenimente din proces, evenimente de la PLC sau de la functiile de monitorizare din WinCC (stari de operare, erori, etc) sunt afisate in forma de mesaje.

In acest scop, mesajele trebuie configurate si impartite in clase de mesaje.

Sistemul de mesaje

- Raportarea cronologica corecta a mesajelor
- Confirmarea mesajelor
- Mesaje singulare sau de grup
- 16 clase de mesaje cu 16 tipuri de mesaje fiecare
- Afisarea utilizand mesaje line-oriented si grafice
- Semnalizare acustica
- Mesajele pot fi selectate
- Informatii mesaj
- Incarcarea permanenta fara pierderi a mesajelor
- Arhivarea pe termen lung, in RT

4.2. Arhivarea

Pentru arhivarea mesajelor, WinCC foloseste arhivarea ciclica (ring buffer) de dimensiune configurabila. Se pot configura cu sau fara backup. Fisierele arhiva sunt stocate intotdeauna pe calculatorul local in proiectul asociat. Arhivarea mesajelor WinCC consta in segmente singulare multiple.

Configurarea sistemului de mesaje cu blocuri de mesaje, mesaje, clase de mesaje, arhive si WinCC Alarm Control

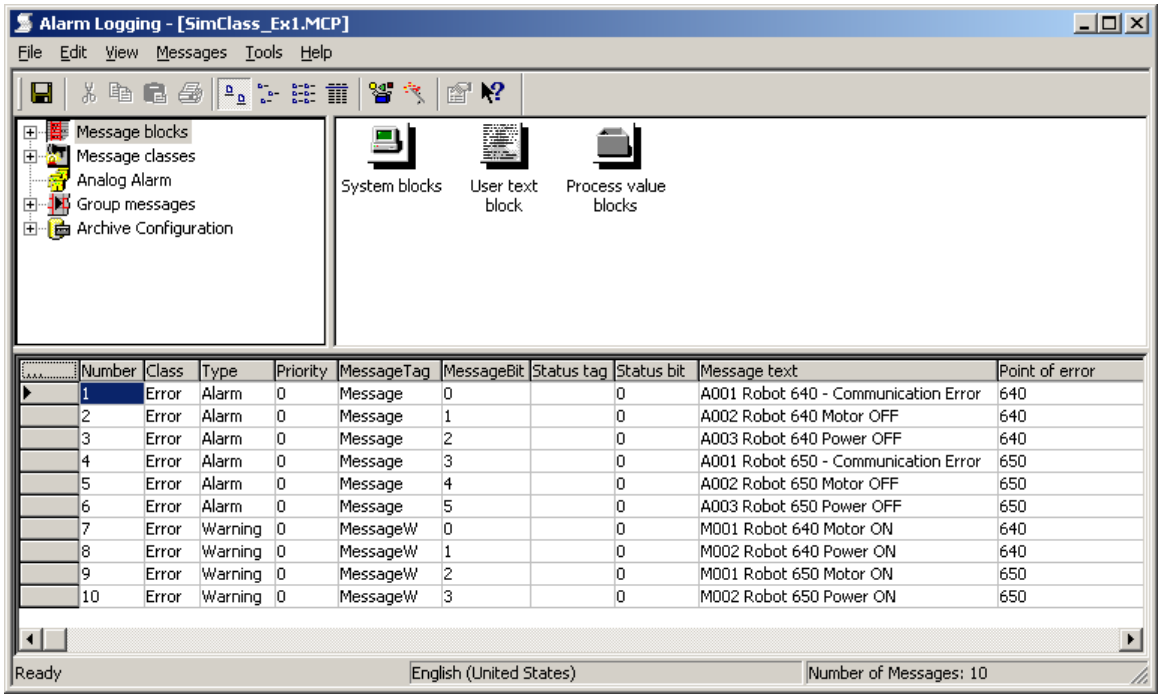
Dupa ce sistemul de mesaje este chemat :

4.3. WinCCExplorer / Alarm Logging

-fereastra din stanga afiseaza editorul

-fereastra din dreapta afiseaza rezultatul configurarii.

Fereastra tabel in partea de jos reprezinta posibilitati de editare pentru mesaje texte, tag-uri etc.



Number	Class	Type	Priority	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error
1	Error	Alarm	0	Message	0		0	A001 Robot 640 - Communication Error	640
2	Error	Alarm	0	Message	1		0	A002 Robot 640 Motor OFF	640
3	Error	Alarm	0	Message	2		0	A003 Robot 640 Power OFF	640
4	Error	Alarm	0	Message	3		0	A001 Robot 650 - Communication Error	650
5	Error	Alarm	0	Message	4		0	A002 Robot 650 Motor OFF	650
6	Error	Alarm	0	Message	5		0	A003 Robot 650 Power OFF	650
7	Error	Warning	0	MessageW	0		0	M001 Robot 640 Motor ON	640
8	Error	Warning	0	MessageW	1		0	M002 Robot 640 Power ON	640
9	Error	Warning	0	MessageW	2		0	M001 Robot 650 Motor ON	650
10	Error	Warning	0	MessageW	3		0	M002 Robot 650 Power ON	650

Fig. 4.1

4.4. Editarea de mesaje bloc

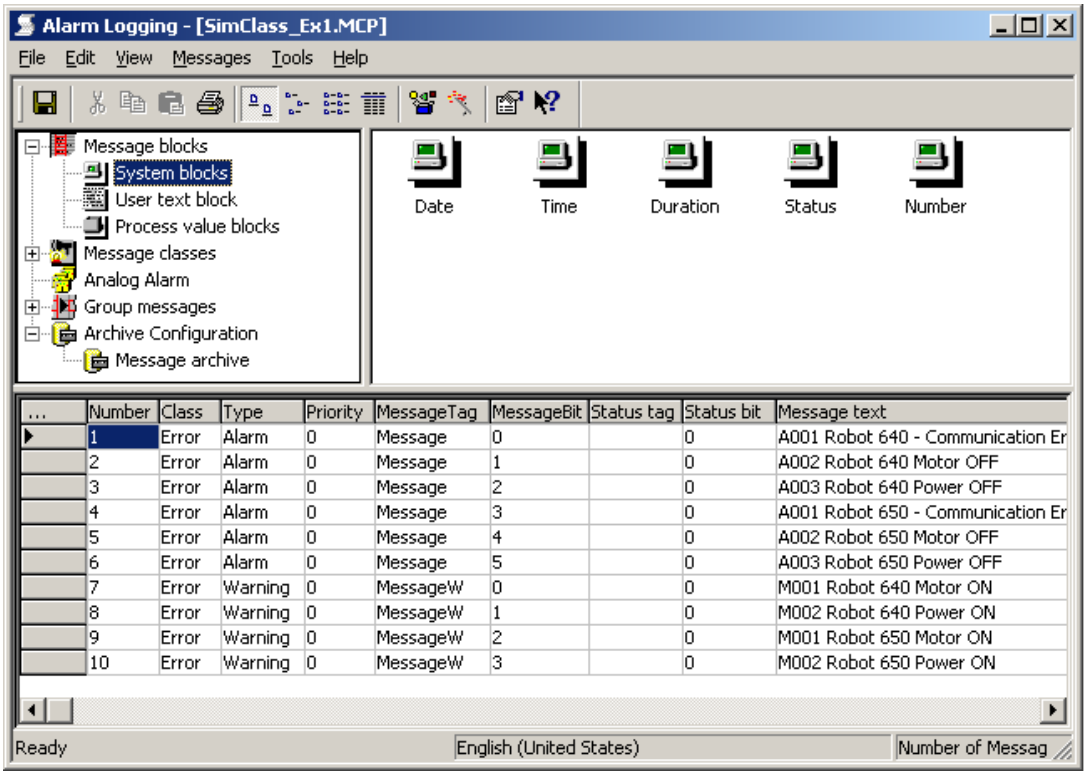

Alarm Logging - [SimClass_Ex1.MCP]
File Edit View Messages Tools Help
Message blocks
System blocks
User text block
Process value blocks
Message classes
Analog Alarm
Group messages
Archive Configuration
Message archive
Date
Time
Duration
Status
Number
... Number Class Type Priority MessageTag MessageBit Status tag Status bit Message text
1 Error Alarm 0 Message 0 0 A001 Robot 640 - Communication Er
2 Error Alarm 0 Message 1 0 A002 Robot 640 Motor OFF
3 Error Alarm 0 Message 2 0 A003 Robot 640 Power OFF
4 Error Alarm 0 Message 3 0 A001 Robot 650 - Communication Er
5 Error Alarm 0 Message 4 0 A002 Robot 650 Motor OFF
6 Error Alarm 0 Message 5 0 A003 Robot 650 Power OFF
7 Error Warning 0 MessageW 0 0 M001 Robot 640 Motor ON
8 Error Warning 0 MessageW 1 0 M002 Robot 640 Power ON
9 Error Warning 0 MessageW 2 0 M001 Robot 650 Motor ON
10 Error Warning 0 MessageW 3 0 M002 Robot 650 Power ON
Ready
English (United States)
Number of Messag

Fig. 4.2

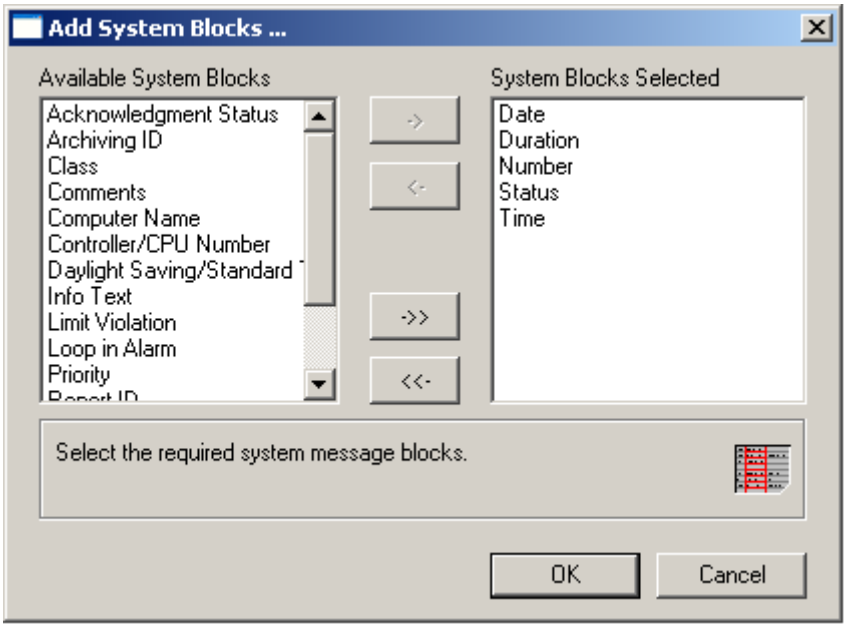


Fig. 4.3

4.5. Structura de mesaje pe clase si tipuri

Clasele de mesaje difera in filosofia de confirmare. Mesajele cu aceeasi filosofie de confirmare se pot grupa impreuna intr-o singura clasa de mesaje. In *Alarm Logging*, clasele de mesaje "*Error*", "*System, requires acknowledgement*" si "*System, without acknowledgement* " sunt preconfigurate. In WinCC se pot defini pana la 16 clase de mesaje.

Tipurile de mesaje sunt subgrupe ale claselor de mesaje si pot diferi sub aspectul tipului de culoare a starii mesajelor. Se pot crea pana la 16 tipuri de mesaje pentru fiecare clasa de mesaje din WinCC.

Exemplu:

1. *Impartire tehnologica*

Clase de mesaje: Turbina 1; Turbina 2

Tipuri: Alarm; Warning; Failure

2. *Impartire pe prioritati*

Clase de mesaje: Alarm; Failure; Error

Tipuri: Cauze electrice; Cauze mecanice

4.6. Editarea si configurarea claselor de mesaje

Editare

Clase de mesaje

1. Din "*Message Classes*"
2. Selecteaza comanda "*Add/Remove...*".
3. Muta clasa de mesaje din partea stanga in partea dreapta.
4. "OK".

Class Properties

1. Selecteaza "*Properties*". Se poate modifica numele (exp. message class 2 -> Turbina1) si se insereaza.

2. "OK"

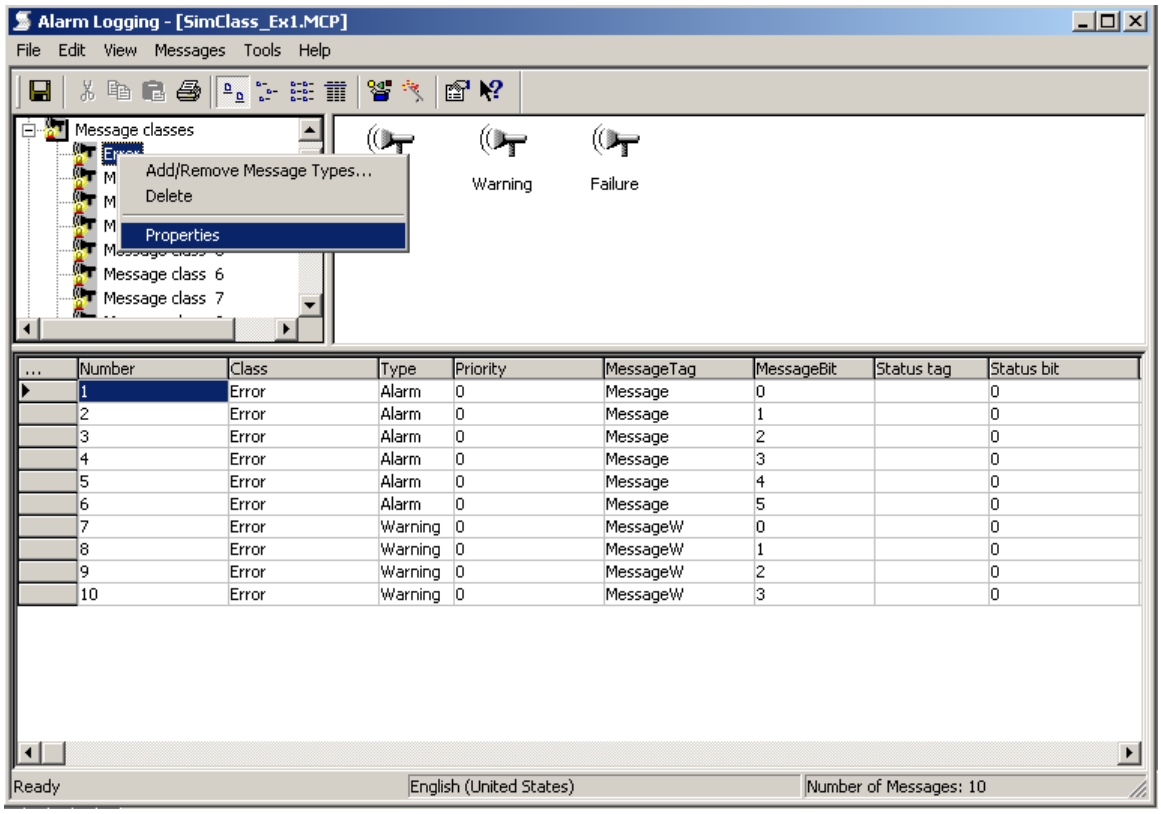

Alarm Logging - [SimClass_Ex1.MCP]
File Edit View Messages Tools Help
Message classes
Add/Remove Message Types...
Delete
Properties
Message class 6
Message class 7
Warning
Failure
... Number Class Type Priority MessageTag MessageBit Status tag Status bit
1 Error Alarm 0 Message 0 0
2 Error Alarm 0 Message 1 0
3 Error Alarm 0 Message 2 0
4 Error Alarm 0 Message 3 0
5 Error Alarm 0 Message 4 0
6 Error Alarm 0 Message 5 0
7 Error Warning 0 MessageW 0 0
8 Error Warning 0 MessageW 1 0
9 Error Warning 0 MessageW 2 0
10 Error Warning 0 MessageW 3 0
Ready
English (United States)
Number of Messages: 10

Fig. 4.4

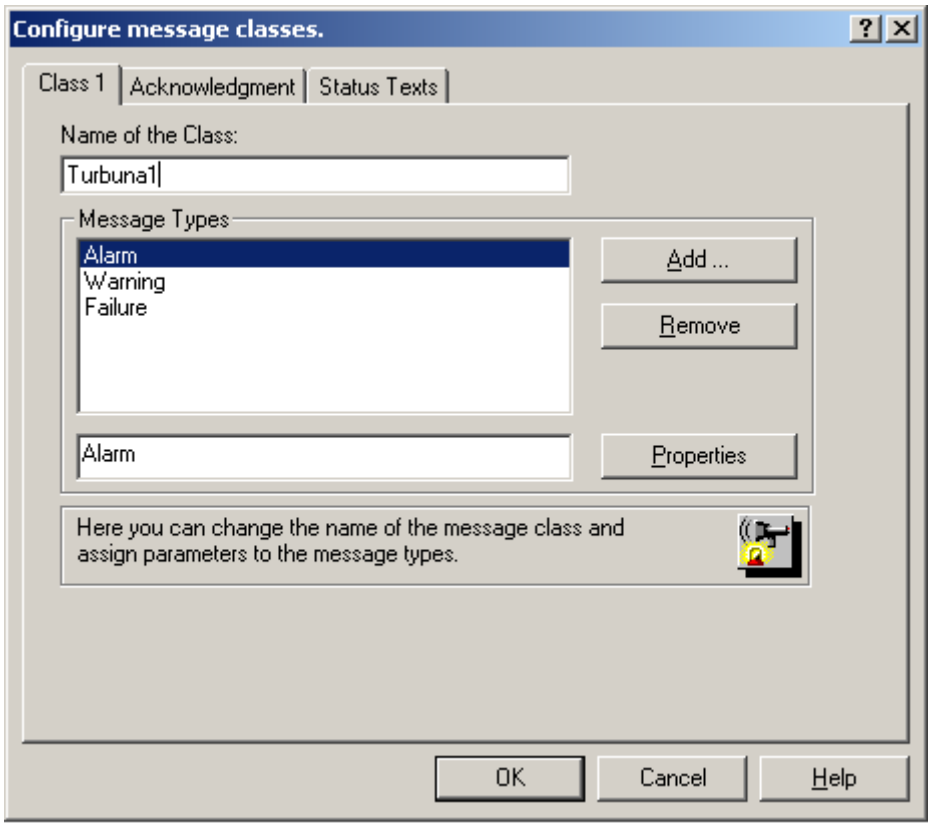


Fig. 4.5

4.7. Editarea unui mesaj

Pentru configurarea unui mesaj se asigneaza un *message tag* si un *message bit*. Mesajele singulare se pot grupa in grupuri de mesaje.

1. Selecteaza un mesaj din tabel.
2. Cheama "*Properties*".

   Se pot selecta parametrii mesajelor
3. "OK"

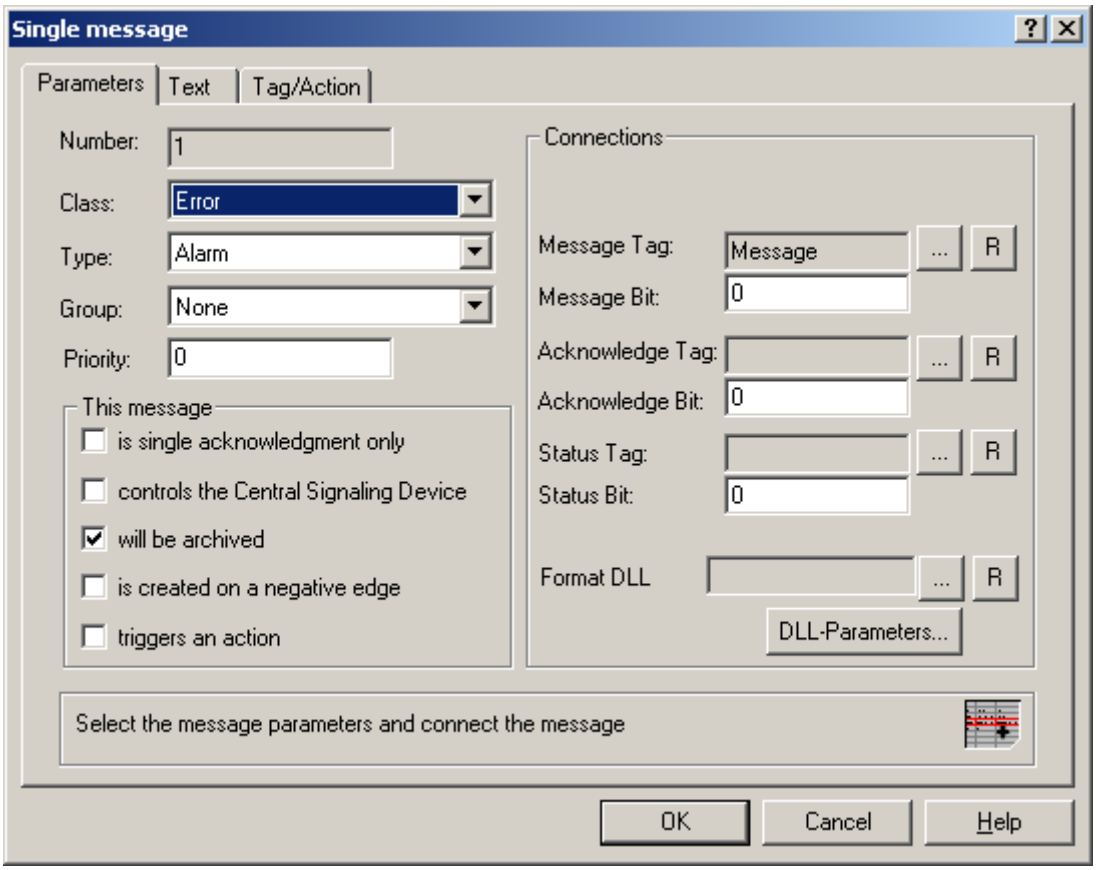


Fig. 4.6

Graphics Designer / WinCC Alarm Control este fereastra in care mesajele sunt afisate la runtime.

Dupa ce se editeaza *Alarm Control*, fereastra *Properties* apare automat.

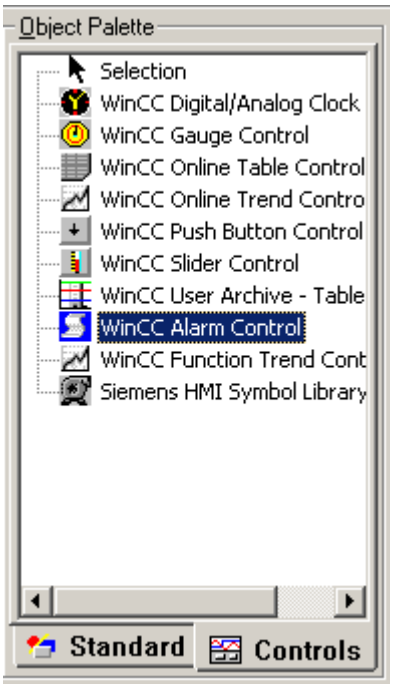

Object Palette
Selection
WinCC Digital/Analog Clock
WinCC Gauge Control
WinCC Online Table Control
WinCC Online Trend Contro
WinCC Push Button Control
WinCC Slider Control
WinCC User Archive - Table
WinCC Alarm Control
WinCC Function Trend Cont
Siemens HMI Symbol Library
Standard
Controls

Fig. 4.7

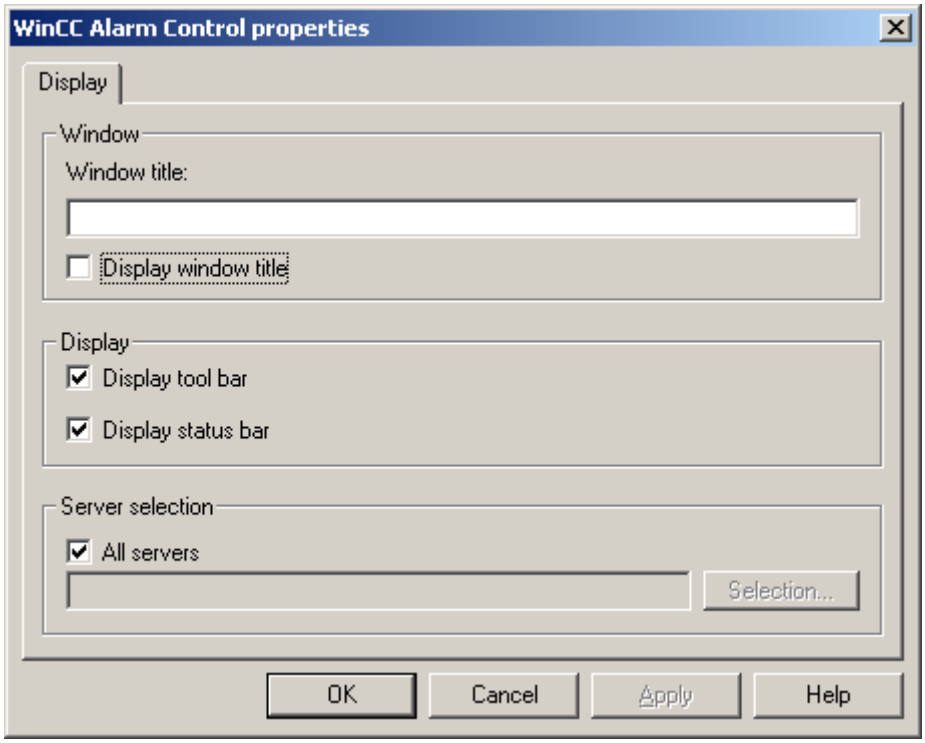

WinCC Alarm Control properties
Display
Window
Window title:
Display window title
Display
Display tool bar
Display status bar
Server selection
All servers
Selection...
OK
Cancel
Apply
Help

Fig. 4.8

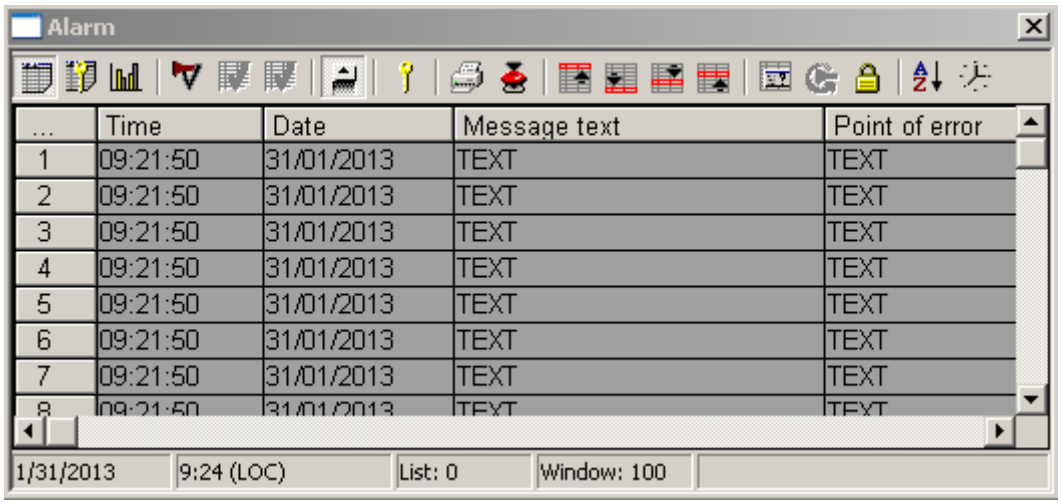


Fig. 4.9

Cu ajutorul ferestrei de proprietati a *WinCC Alarm Control*, se stabileste proiectarea si afisarea ferestrei de mesaje. In fereastra *Properties* a elementului *WinCC Alarm* se pot asigna parametri (*message lists*, *message blocks*, etc).

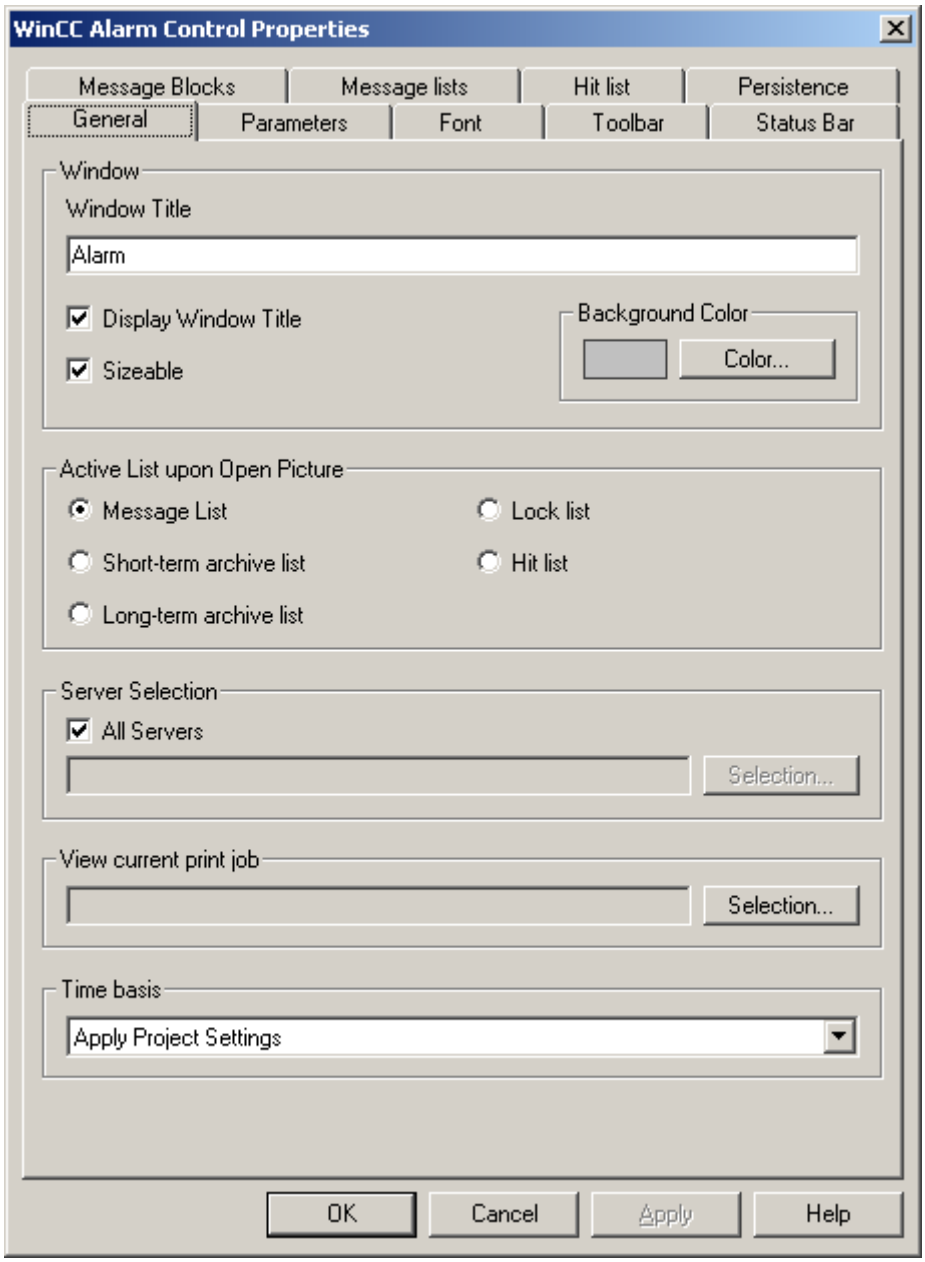

WinCC Alarm Control Properties
Message Blocks
Message lists
Hit list
Persistence
General
Parameters
Font
Toolbar
Status Bar
Window
Window Title
Alarm
Display Window Title
Sizeable
Background Color
Color...
Active List upon Open Picture
Message List
Lock list
Short-term archive list
Hit list
Long-term archive list
Server Selection
All Servers
Selection...
View current print job
Selection...
Time basis
Apply Project Settings
OK
Cancel
Apply
Help

Fig. 4.10

Exercitiul 1:

Creaza blocurile de mesaje specificate:

System Blocks:

1. Date (Properties: 4 digit)
2. Time (Properties: Time format)
3. Status (Properties: 2 characters)

User text block:

1. Message text (Properties: 25 characters)
2. Point of error (Properties: 20 characters)

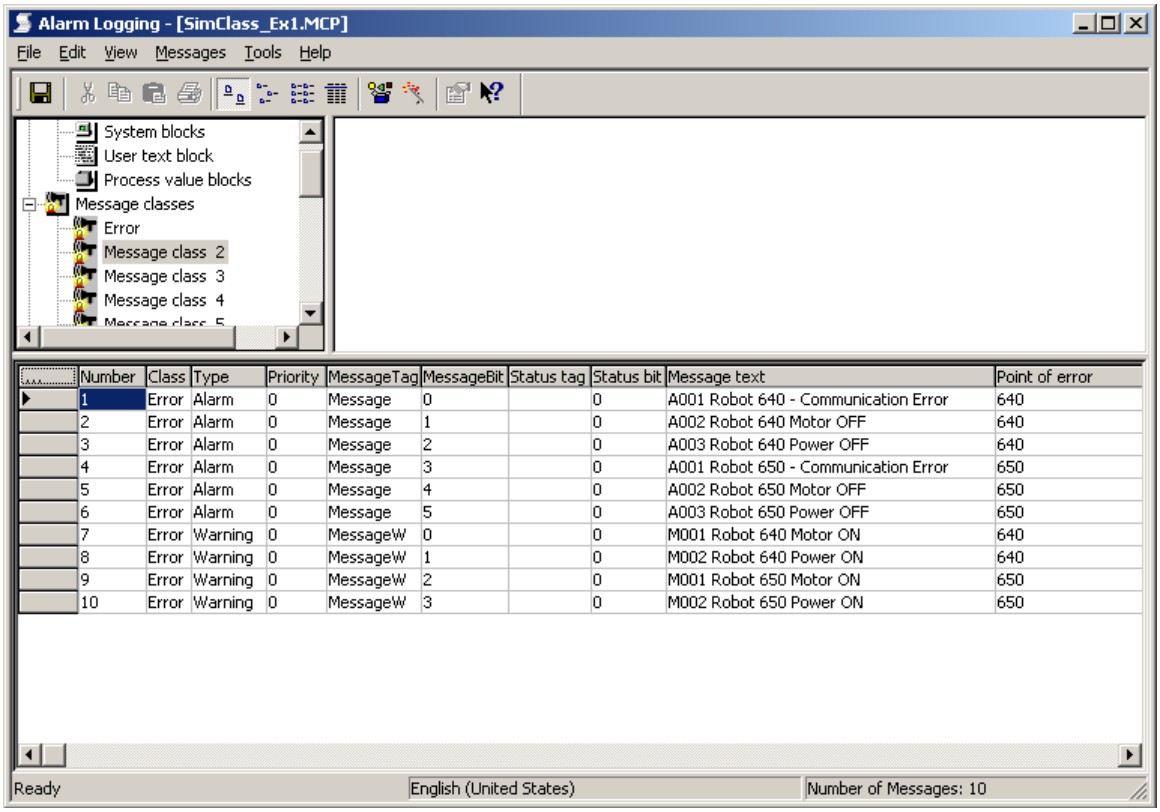
Exercitiul 2:

1. Creaza 10 mesaje cu parametrii specificati in figura de mai jos.

Type: Alarm
A001 Robot 640 - Communication Error
A002 Robot 640 Motor OFF
A003 Robot 640 Power OFF
A001 Robot 650 - Communication Error
A002 Robot 650 Motor OFF
A003 Robot 650 Power OFF

Type: Warning

M001 Robot 640 Motor ON
M002 Robot 640 Power ON
M001 Robot 650 Motor ON
M002 Robot 650 Power ON



Number	Class	Type	Priority	MessageTag	MessageBit	Status tag	Status bit	Message text	Point of error
1	Error	Alarm	0	Message	0		0	A001 Robot 640 - Communication Error	640
2	Error	Alarm	0	Message	1		0	A002 Robot 640 Motor OFF	640
3	Error	Alarm	0	Message	2		0	A003 Robot 640 Power OFF	640
4	Error	Alarm	0	Message	3		0	A001 Robot 650 - Communication Error	650
5	Error	Alarm	0	Message	4		0	A002 Robot 650 Motor OFF	650
6	Error	Alarm	0	Message	5		0	A003 Robot 650 Power OFF	650
7	Error	Warning	0	MessageW	0		0	M001 Robot 640 Motor ON	640
8	Error	Warning	0	MessageW	1		0	M002 Robot 640 Power ON	640
9	Error	Warning	0	MessageW	2		0	M001 Robot 650 Motor ON	650
10	Error	Warning	0	MessageW	3		0	M002 Robot 650 Power ON	650

Fig. 4.11

2. Creaza un ecran Alarme.pdl. Insereaza un element WinCC Alarm control.

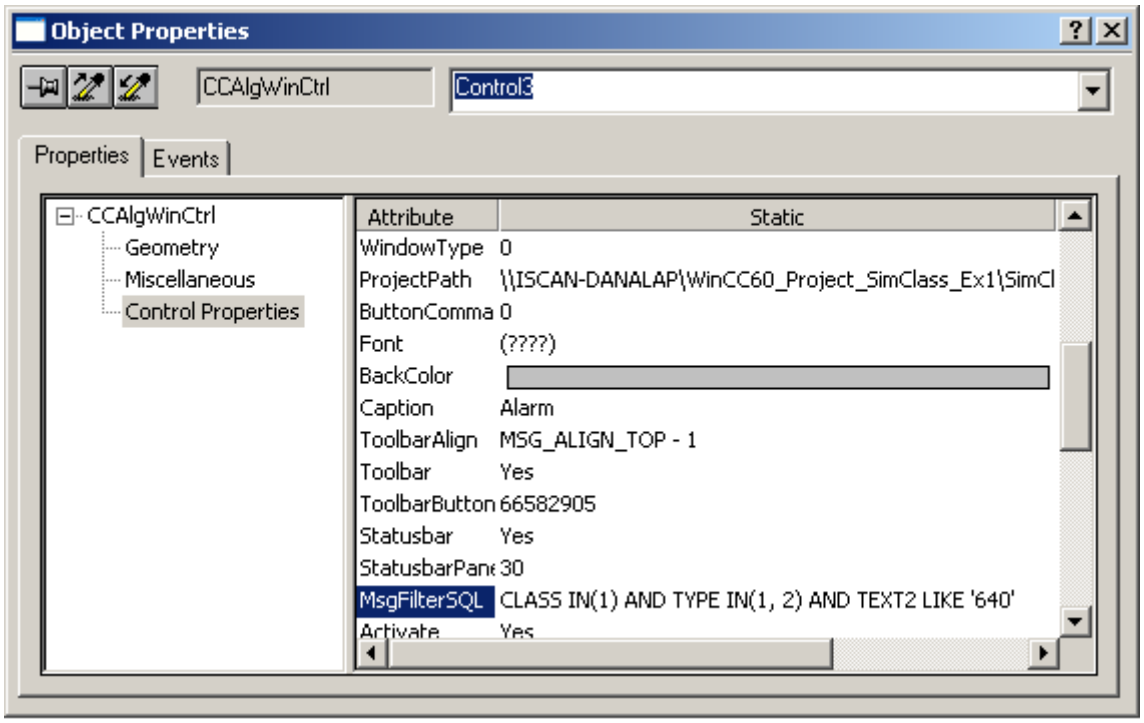


Fig. 4.12

La *Properties/MsgFilterSQL* se face o filtrare dupa Point of Error ‘640’:

CLASS IN(1) AND TYPE IN(1, 2) AND TEXT2 LIKE '640'

Pentru test insereaza un element *Check Box* (*Properties / Output /Input/Tag: MessageW*).

Insereaza si configureaza un buton “Short Archive” (*Events/Mouse Action/C-Action*)

```
#include "apdefap.h"
void      OnClick(char*      lpszPictureName,      char*
lpszObjectName, char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
```

```
// next PicID : 1
// WINCC:PICNAME_SECTION_END

AXC_OnBtnArcShort(lpszPictureName,"Control1");
     //Return-Type: BOOL
}
```

Insereaza si configureaza un buton “Long Archive” (*Events/Mouse Action/C-Action*)

```
#include "apdefap.h"
void      OnClick(char*      lpszPictureName,      char*
lpszObjectName, char* lpszPropertyName)
{
// WINCC:TAGNAME_SECTION_START
// syntax: #define TagNameInAction "DMTagName"
// next TagID : 1
// WINCC:TAGNAME_SECTION_END

// WINCC:PICNAME_SECTION_START
// syntax: #define PicNameInAction "PictureName"
// next PicID : 1
// WINCC:PICNAME_SECTION_END

AXC_OnBtnArcLong(lpszPictureName,"Control1");
     //Return-Type: BOOL
}
```

Capitolul 5. Sisteme SCADA realizate cu automate programabile

Sistemele SCADA actuale sunt sisteme *distribuite si ierarhizate.* Echipamentele RTU (Remote Terminal Unit) sunt statii de lucru locale, care de fapt sunt echipamente electronice inteligente IED (Intelligent Electronic Device).

- Echipamentele RTU din structurile SCADA distribuite si ierarhizate pot fi *automate programabile* configurate hardware corespunzator nivelului de complexitate cerut de aplicatie.

Automatele programabile sunt structurate pe module, ceea ce permite o configurare hardware flexibila.

Modularizarea a fost un principiu de succes care a permis reducerea timpilor de configurare si de punere in functiune a instalatiilor. Controlul distribuit a permis distributia inteligentei atat pe orizontala cat si pe verticala si prezenta ei exact la locul unde este necesara.

STEP 7 reprezinta softul de programare pentru SIMATIC S7, deci si pentru S7-300. Step 7 ofera intregul suport pentru configurare, programare, atribuirea parametrilor pentru S7–300. Acest soft ofera un sprijin efectiv in rezolvarea unei probleme date.

Softul se porneste foarte usor printr-un dublu clic pe iconita de Simatic Manager.

Rezultat: Se va deschide fereastra pentru un nou proiect al lui Simatic Manager.

Fig.5.1. Interfata softului de programare

Simatic Manager este interfata de programare cu STEP 7. Acesta ne ofera o reprezentare ierarhica a tuturor obiectelor dintr-un proiect, facand astfel posibil accesul la toate functiile necesare rezolvarii unui task in automatizare.

Pornind de la Simatic Manager putem:

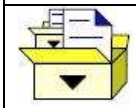
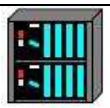
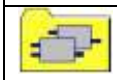
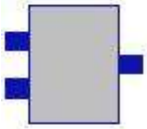
- Configura si atribui parametrilor lui S7-300.
- Programa automatul S7-300.

Pentru a putea oferii solutii de automatizare trebuiesc stabilite urmatoarele informatii de baza:

- Ce obiecte ne sunt necesare pentru a lucra cu Step 7?
- Cum se poate lucra cu aceste obiecte?

In tabelul 5.1. sunt prezentate obiectele cu care lucreaza Step 7.

Tabelul 5.1.

Simbol	*Obiect*	*Descriere*	*Subordonat*
	Proiect	Intarea tuturor datelor si programelor a unei solutii automate.	Se gaseste in varful ierarhiei de obiecte.
	Statie 300 Simatic	O instalatie hard cu unul sau mai multe module programabile.	Proiect
	Modul Programabil	Un modul programabil (CPU).	Statie 300 Simatic
	Program S7	Contine blocuri si fisiere sursa.	Modul programabil sau proiect.
	Blocuri	Contine programele folosite de utilizator.	Programe S7
	Bloc (offline) Bloc (online)	Pot fii spre exemplu blocuri logice (OB si FC)	Blocuri

Un proiect contine intreaga baza de date pentru solutia de automatizare a unui task. Crearea unui proiect sau a structurii proiectului este un lucru esential in folosirea lui Step 7. Figura urmatoare indica structura unui proiect:

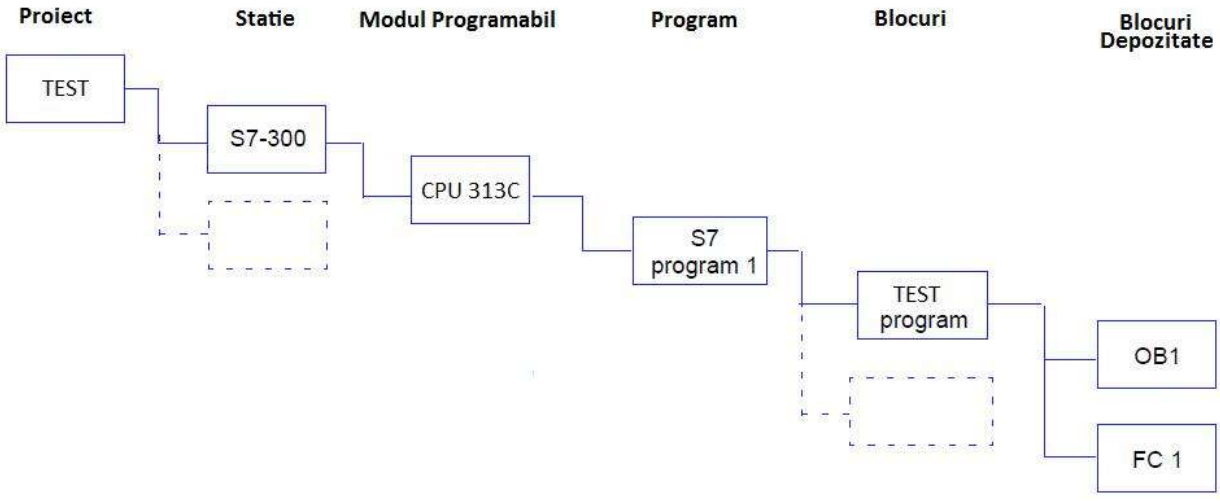


Fig.5.2. Structura ierarhica a unui proiect

5.1. Crearea unui nou proiect

Deschideti Simatic Manager apasand dublu click pe iconita de pe Desktop.

1. Din meniu programului dati clic pe comanda **FILE -> New -> Project** in Simatic Manager sau pe iconita reprezentativa din bara de unelte.

2. In urmatoarea casuta de dialog introduceti numele proiectului, in cazul nostru “Proiect TEST” si dati clic pe OK pentru crearea unui nou proiect. Nu uitati sa salvati noul proiect

3. In folderul Student de pe Desktop

Fig.5.3.Crearea unui proiect

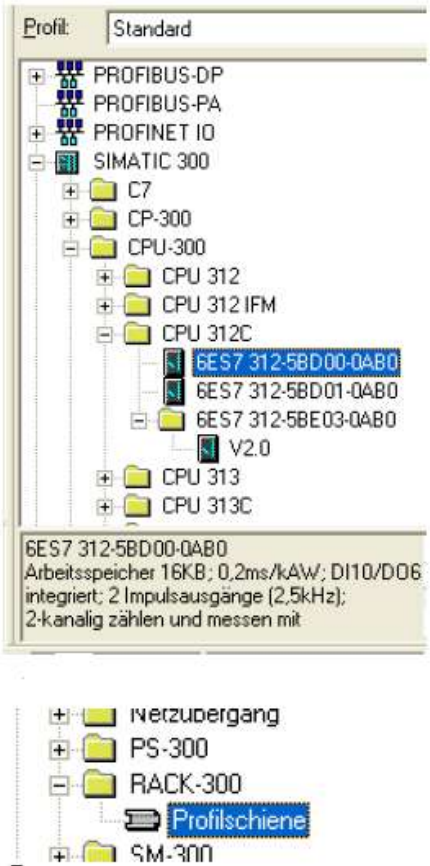


Fig.5.4.Configurarea hardware -

4. Adaugarea unei statii la un proiect existent, deschis, serealizeaza fie din meniu, prin comanda **Insert->Hardware-> Simatic 300 Station**, fie printr-un clic pe semnul "+" din fata proiectului daca statia nu este afisata.

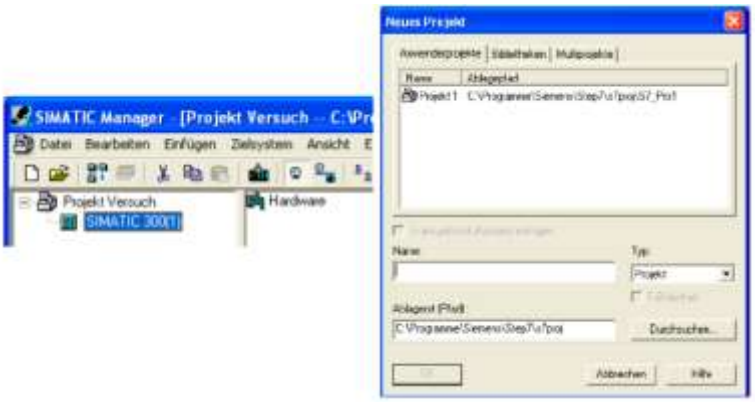


Fig.5.5.Introducerea unei statii Simatic 300

5.2. Configurarea hardware

Configurarea reprezinta aranjarea modulelor intr-un tabel.

Modulele se aranjeaza pe un stativ folosind STEP 7, exact ca in sistemul real. Modulele se selecteaza dintr-un catalog electronic si apoi se introduc in pozitia corespunzatoare din tabelul de configurare.

Stativul din tabelul de configurare trebuie sa corespunda celui real. Step 7 atribuie adrese fiecarui modul din tabelul de configurare.

Atribuirea presupune stabilirea caracteristicilor si comportamentului modulelor.

Inainte de a introduce o noua configurare sau atribuire de parametrii trebuie indeplinite doua cerinte: crearea unui nou proiect apoi selectarea obiectului pentru care se doreste configurarea.

Anterior descarcarii noii configuratii, trebuie mai intai efectuata o resetare a CPU-ului, pentru a ne asigura ca nu exista alte blocuri vechi in memoria acestuia.

Procedura de baza pentru configurarea si atribuirea parametrilor este prezentata in figura de mai jos:

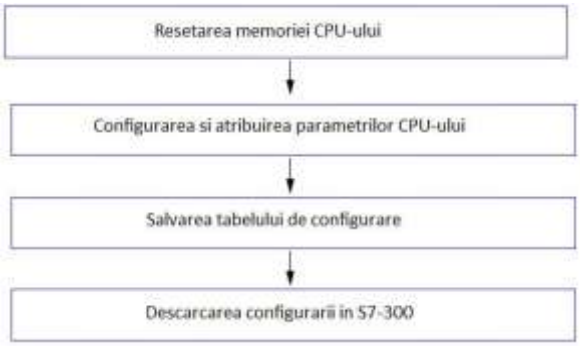

Resetarea memoriei CPU-ului
Configurarea si atribuirea parametrilor CPU-ului
Salvarea tabelului de configurare
Descarcarea configurarii in S7-300

Fig.5.6.Configurarea hardware - 2

In continuare se vor indica pasii ce urmeaza sa fie parcursi pentru a configura si atribuii parametrii lui S7-300.

- Selectati statia SIMATIC 300 din proiectul vostru si intrati in tabelul de configurare cu comanda **Edit-> Open Object** (din meniu). Rezultatul aceste operatii este aparitia unui ecran cu o fereastra de lucru si un calalog hardware ce contine toate modulele disponibile. In cazul in care catalogul nu este afisat, il puteti deschide folosind combinatia de taste "**Ctrl+K**".
- In catalogul hardware selectati stativul de montare astfel: **Simatic 300-> Rack-> Rail** si pozitionati-l in fereastra de lucru. Rezultatul: aparitia unui tabel ce afiseata sina si sloturile sale.
- Dati clic pe tabel si selectati vederea detaliata a tabelului de configurare cu ajutorul comenzii **View-> Details**. Ca rezultat va fi aparitia unui tabel de configurare cu o vedere de ansamblu.
- Din catalogul hardware selectati modulele folosite cu Simatic 300.
  - ✓ Sursa de tensiune (PS 307 2A)
  - ✓ CPU (313C)

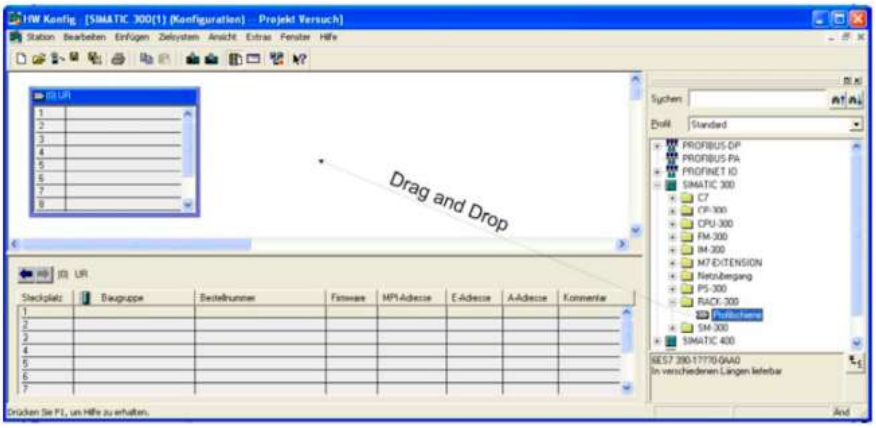


Fig.5.7.Configurarea hardware - 3

Rezulatul aceste operatii va fi apariatia modulelor selectate in liniile corespunzatoare tabelului de configurare.

Obs: Tabelul de configurare trebuie se fie completat in conformitate cu configurarea hardware a lui S7-300. Pozitia corespunzatoare in tabelul de configurare este data de numerotarea modulelor pe sina.

In figura urmatoare este prezentat un exemplu de tabel de configurare:

Fig.5.8.Configurarea hardware - 4

5.3. Procedura de programare a blocurilor

Programarea automatului se realizeaza foarte usor prin crearea unui program care se descarca in CPU-ul lui S7-300. Programul creat de utilizator este format din mai multe blocuri care ofera posibilitate de structurare a acestuia.

OB – blocuri organizationale – reprezinta o interfata inte sistemul de operare al CPU-ului si programul utilizatorului. Ordinea in care este executat un program este specificata in OB.

FC – bloc functie – este un bloc logic fara memorie. Acest bloc este folosit in special pentru crearea de functii folosite uzual in program.

Procedura de baza pentru programarea blocurilor este urmatoarea:

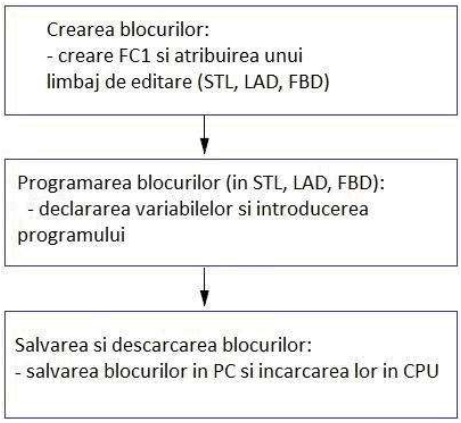

Crearea blocurilor:
- creare FC1 si atribuirea unui
limbaj de editare (STL, LAD, FBD)
Programarea blocurilor (in STL, LAD, FBD):
- declararea variabilelor si introducerea
programului
Salvarea si descarcarea blocurilor:
- salvarea blocurilor in PC si incarcarea lor in CPU

Fig.5.9. Procedura de baza pentru programarea blocurilor

Crearea FC-urilor si OB-urilor.

In continuare se vor indica pasii ce urmeaza sa fie parcursi pentru crearea FC-urilor si Ob-urilor.

- Deschideti proiectul de test din Simatic Manager cu comanda **File-> Open Project nume.**

Rezultat: va aparea fereastra pentru proiectul de test care va fi deschis offline.

- Deschideti continutul programului de test pana la ultimul nivel prin apasarea pe "+" si selectati "**Blocks**".

Rezultat: OB1 stocat in blocul principal va fi afisat offline.

- Inserati o functie FC1 cu ajutorul comenzii **Insert-> S7 Block-> FC** (din meniu). In casuta de dialog care va aparea selectati limbajul de programare (STL, LAD, FBD) apoi apasati OK.

Rezultat: FC1 si OB1 sunt afisate offline in fereastra proiect.

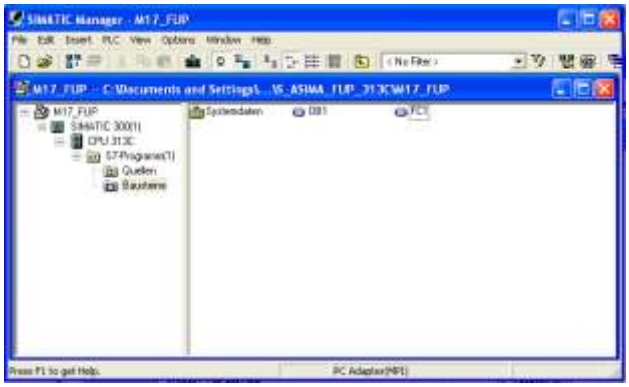


Fig.5.10.

STL = Statement list = este un limbaj de programare textual in Step 7. Sintaxa acestui limbaj este apropiata de limbajul cod masina.Instructiunile si operatiile sunt urmate de adresele corespunzatoare.

LAD = Ladder Logic = este un limbaj de programare grafic in Step 7. Sintaxa acestui limbaj este asemanatoare unei diagrame, permitand astfel o urmarire mai usoara a fluxului de curent.

FBD = Function Block Diagram = este de asemeni un limbaj grafic de programare in Step 7. Sintaxa este reprezentata de blocuri logice similare cu cele din algebra booleana.

STL, LAD, FBD sunt integrate in softul standard al lui Step 7. Astfel, dupa instalarea lui Step 7, toate editoarele, compilatoarele si functiile de test pentru STL, LAD, FBD sunt disponibile.

Bibliografie

1. Daniela Hossu, Ioana Fagarasan, Andrei Hossu, *Proiectarea aplicatiilor SCADA. Studii de caz*, Ed. Printech, ISBN 978-606-521-973-1, Bucuresti, 2013.

2. Sergiu Stelian Iliescu, Dan Mihoc, *Automatizari si protectii prin relee in sistemele electroenergetice*, E.D.P. Bucuresti, 1983.

3. R. Dobrescu, M. Dobrescu, Henri-George Coanda, *Aplicatii distribuite*, Ed. Bibliotheca, Targoviste, 2003.

4. Daniela Hossu, *Sisteme de transmitere a datelor cu aplicatii in energetica*, Ed. Printech, 2007.

5. Daniela Hossu, *Infrastructuri de comunicatie pentru sisteme evoluate de control*, Ed. Printech, ISBN 978-973-718-918-9, Bucuresti, 2008.

6. Dobrescu R., D. Andone, M. Dobrescu, *Dedicated Software for Fieldbus Control Applications*, Proc. Of the European Control Conference, ECC01, Porto, Portugal, Sept 3-7, 2001, pp.2273-2278, poster paper.

7. Dobrescu R., G. Ionescu, D. Popescu, D. Andone, M. Dobrescu, G. Florea, *A New Process Control Strategy Based on Fieldbus*, LSS 2001, Bucharest, Romania, July 18-20, 2001, pp. 208-214.

8. *Supervisory Control and Data Acquisition (SCADA) Systems*, NCS TIB 04-1, Communication Technologies, Inc.14151 Newbrook Drive, Suite 400 Chantilly, Virginia 20151, October 2004.

9. Peter King, *SCADA Systems –Looking Ahead*, Control Microsystems White Paper August 2005.

10. Dennis Gaushel, Henry Darlington, *Supervisory Control and Data Acquisition*, PROCEEDINGS OF THE IEEE, VOL. 75, NO. 12, DECEMBER 1987

11. Norikazu Toshida, Mario Uesugi, Yuji Nakata, Masaaki Nomoto, Tomonobu Uchida, *Open Distributed EMS/SCADA System*, Hitachi Review Vol. 47 (1998), No. 5.

12. Jurgen Prange and Judith Bishop, *Distributed Operating Systems A Study In Applicability*, SART / SACJ, No 24, 1999.

13. Daneels A., Salter W., *WHAT IS SCADA?*, International Conference on Accelerator and Large Experimental Physics Control Systems, 1999, Trieste, Italy.

14. Arghira Nicoleta, Hossu, Daniela, Făgărășan, Ioana, Iliescu, S. St., Costianu, D.R., Modern SCADA philosophy in power system operation - A survey, Scientific Bulletin, University POLITEHNICA Bucharest, Series C: Electrical Engineering, vol. 73, nr. 2, ISSN 1454234x, pp. 153-166, 2011

Investește în oameni !
Proiect cofinanțat din FONDUL SOCIAL EUROPEAN
prin Programul Operațional Sectorial Dezvoltarea Resurselor Umane 2007 – 2013
Axa prioritară 1: „EDUCAȚIA ȘI FORMAREA PROFESIONALĂ ÎN SPIJINUL CREȘTERII ECONOMICE ȘI DEZVOLTĂRII SOCIETĂȚII BAZATE PE CUNOAȘTERE"
Domeniul major de intervenție 1.2 „Calitate în învățământul superior"
Cerere de propuneri de proiecte: nr. 86 „Universitate pentru viitor"
Titlul proiectului: ***"Rețea națională de centre pentru dezvoltarea programelor de studii cu rute flexibile și a unor instrumente didactice la specializarea de licență și masterat, din domeniul Ingineria Sistemelor"***
Numărul de identificare al contractului: **POSDRU/86/1.2/S/63806**

Obiectivul general al proiectului – vizează "asigurarea unui învățământ superior de calitate și performant, în domeniul fundamental de studiu "Ingineria Sistemelor" – IS, în vederea consolidării unei gândiri structurate și flexibile, ameliorării rezultatelor învățării la nivel de sistem, dobândirii de cunoștințe și competențe adaptate cerințelor pieței muncii, prin restructurarea conținuturilor programelor de studii din domeniu, și prin conștientizarea factorilor economici de utilitatea sprijinirii inserției pe piața muncii a acestor absolvenți."

Obiective specifice (OS)
OS 1 - "Identificarea necorelărilor dintre actuala curriculă și cerințele pieței muncii", sprijină intențiile promotorului proiectului de a constata care sunt eventualele necorelări între cerințele pieței muncii și oferta furnizorilor de educație în domeniul IS, în scopul identificării unor soluții și elaborării unor măsuri de implementare a acestora.
OS 2 – "Crearea unui plan de învățământ cu ajutorul unui Ghid curricular IS care va asigura formarea unor competențe compatibilizate la nivel european ale viitorilor specialiști IS", corespunde obiectivelor POS DRU prin elaborarea și adaptarea programelor de studiu pentru licență ăi masterat în domeniul IS cu ajutorul unor curricule îmbunătățite, în conformitate cu CNCIS și cu cerințele pieței muncii.
OS 3 - "Creșterea calității pregătirii studenților de la specializările IS în raport cu cerințele potențialilor angajatori, prin elaborarea și transpunerea în practica a unui proiect de curricula îmbunătățită".
OS 4 - "Promovarea parteneriatului între instituțiile de învățământ superior și actorii cheie relevanți în domeniul IS" - conduce la satisfacerea priorităților acestei cereri de propuneri de proiecte prin dezvoltarea și consolidarea unei rețele între universități, pe de o parte și actori cheie, relevanți pe piața muncii din domeniul IS (angajatori, patronate, asociații profesionale), pe de altă parte.
OS 5 – "Crearea unei rețele de 5 centre pentru cadrele didactice și studenții implicați în creșterea calității învățământului în Ingineria Sistemelor în România și în desfășurarea de bune practici la nivel național. "
OS 6 – "Realizarea unui portal *online* ca sursă de informare și documentare pentru profesorii și studenții care doresc, cât și ca un instrument modern și flexibil de comunicare, învățare și evaluare, stimularea cooperării și comunicării între studenți și chiar între studenți și elevii din licee în scopul atragerii acestora către domeniul IS"

UNIUNEA EUROPEANĂ | GUVERNUL ROMÂNIEI MINISTERUL MUNCII, FAMILIEI, PROTECTIEI SOCIALE ŞI PERSOANELOR VÂRSTNICE AMPOSDRU | Fondul Social European POSDRU 2007-2013 | Instrumente Structurale 2007-2013 | MINISTERUL EDUCAŢIEI NAŢIONALE OIPOSDRU | UNIVERSITATEA TEHNICĂ DE CONSTRUCŢII DIN BUCUREŞTI

Investeşte în oameni !

Proiect cofinantat din FONDUL SOCIAL EUROPEAN prin Programul Operaţional Sectorial Dezvoltarea Resurselor Umane 2007 – 2013

Axa prioritară 1: „EDUCAŢIA ŞI FORMAREA PROFESIONALĂ ÎN SPRIJINUL CREŞTERII ECONOMICE ŞI DEZVOLTĂRII SOCIETĂŢII BAZATE PE CUNOAŞTERE"
Domeniul major de intervenţie 1.2 „Calitate în învăţământul superior"
Cerere de propuneri de proiecte: nr. 86 „Universitate pentru viitor"

Titlul proiectului:

Reţea naţională de centre pentru dezvoltarea programelor de studii cu rute flexibile şi a unor instrumente didactice la specializarea de licenţă şi masterat, din domeniul Ingineria Sistemelor.

Numărul de identificare al contractului: **POSDRU/86/1.2/S/63806**
Beneficiar: Universitatea Tehnică de Construcţii Bucureşti

Manager de proiect: prof. univ. dr. Mariana Marinescu

Editat de Beneficiar – Universitatea Tehnică de Construcţii Bucureşti,
Editura CONSPRESS

Publicat la data : 07.06.2013

Conţinutul acestui material nu reprezintă în mod obligatoriu poziţia oficială a Uniunii Europene sau a Guvernului României

Proiect cofinantat din Fondul Social European prin Programul Operational Sectorial Dezvoltarea Resurselor Umane 2007-2013 – Investeşte în OAMENI | MINISTERUL EDUCAŢIEI NAŢIONALE – MINISTERUL EDUCAŢIEI NATIONALE | UNIVERSITATEA "POLITEHNICA" DIN BUCUREŞTI | UNIVERSITATEA TEHNICĂ DIN CLUJ-NAPOCA | UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" DIN IAŞI | UNIVERSITATEA "POLITEHNICA" DIN TIMIŞOARA | SC ASTI AUTOMATION SRL

ISBN 978-973-100-281-1